Master's thesis

Master's Programme in Computer Science

How industrial automation systems met the Internet – on SCADA communication protocols and security

Kristian Krok

July 4, 2024

Faculty of Science

University of Helsinki

Contact information

P. O. Box 68 (Pietari Kalmin katu 5)
00014 University of Helsinki,Finland

Email address: info@cs.helsinki.fi
URL: http://www.cs.helsinki.fi/

HELSINGIN YLIOPISTO – HELSINGFORS UNIVERSITET – UNIVERSITY OF HELSINKI

Tiedekunta — Fakultet — Faculty	Koulutusohjelma — Utbildningsprogram — Study programme
Faculty of Science	Master's Programme in Computer Science

Tekijä — Författare — Author

Kristian Krok

Työn nimi — Arbetets titel — Title

How industrial automation systems met the Internet – on SCADA communication protocols and security

Ohjaajat — Handledare — Supervisors

Prof. Valtteri Niemi

Työn laji — Arbetets art — Level	Aika — Datum — Month and year	Sivumäärä — Sidoantal — Number of pages
Master's thesis	July 4, 2024	45 pages

Tiivistelmä — Referat — Abstract

This thesis discusses supervisory control and data acquisition (SCADA) systems and their communication security. SCADA is an ubiquitous framework used in modern industrial control systems for monitoring and controlling operations technology (OT) equipment. This thesis first briefly covers the history and evolution of SCADA, its modern day applications and most common security vulnerabilities. The rest of the thesis discusses SCADA communication and its security aspects.

This thesis goes over the requirements of SCADA system communication and its related security aspects. A more detailed look is taken into SCADA communication protocols and communication security. Additionally some related standards and best practices are introduced. Cyber attacks targeting SCADA systems can have far-reaching real world consequences as shown by several prior known incidents. For this reason research focusing on SCADA communication security is becoming more and more crucial.

ACM Computing Classification System (CCS)
Security and privacy → Network security
Networks → Network protocols
Security and privacy → Systems security

Avainsanat — Nyckelord — Keywords

SCADA systems, communication security, SCADA protocols

Säilytyspaikka — Förvaringsställe — Where deposited

Helsinki University Library

Muita tietoja — övriga uppgifter — Additional information

Networking study track

HELSINGIN YLIOPISTO – HELSINGFORS UNIVERSITET – UNIVERSITY OF HELSINKI

Tiedekunta — Fakultet — Faculty	Koulutusohjelma — Utbildningsprogram — Study programme
Faculty of Science	Master's Programme in Computer Science

Tekijä — Författare — Author

Kristian Krok

Työn nimi — Arbetets titel — Title

How industrial automation systems met the Internet – on SCADA communication protocols and security

Ohjaajat — Handledare — Supervisors

Prof. Valtteri Niemi

Työn laji — Arbetets art — Level	Aika — Datum — Month and year	Sivumäärä — Sidoantal — Number of pages
Master's thesis	July 4, 2024	45 pages

Tiivistelmä — Referat — Abstract

Tämä tutkielma tarkastelee SCADA-järjestelmiä (supervisory control and data acquisition) ja niiden viestintäturvallisuutta. SCADA-järjestelmät ovat erittäin laajasti käytettyjä teollisuusautomaatiossa. Niitä käytetään erilaisten teollisuuden automaatioprosessien (operational technology) laitteiden valvonta- ja ohjaustehtäviin. Tässä tutkielmassa käydään ensin lyhyesti läpi SCADA-järjestelmien historiaa, moderneja käyttökohteita sekä niihin kohdistuvia yleisimpiä turvallisuusuhkia. Tämän jälkeen keskitytään tarkastelemaan SCADA-järjestelmien tiedonvälitystä sekä tiedonvälityksen turvallista toteuttamista.

Tässä tutkielmassa käydään läpi SCADA-järjestelmien tiedonvälitykseen ja sen turvallisuuteen vaikuttavia tekijöitä. Erityisesti tarkastalleen SCADA-järjestelmissä käytettäviä liikenneprotokollia ja protokollaturvallisuutta. Lisäksi esitellään joukko olennaisia standardeja sekä hyviä käytänteitä. SCADA-järjestelmiin kohdistuvilla kyberhyökkäyksillä voi olla kauaskantoisia ja konkreettisia seurauksia. SCADA-järjestelmien turvallisuutta käsittelevän tutkimuksen rooli on noussut tänä päivänä entistä keskeisemmäksi.

ACM Computing Classification System (CCS)
Security and privacy → Network security
Networks → Network protocols
Security and privacy → Systems security

Avainsanat — Nyckelord — Keywords

SCADA systems, communication security, SCADA protocols

Säilytyspaikka — Förvaringsställe — Where deposited

Helsinki University Library

Muita tietoja — övriga uppgifter — Additional information

Networking study track

Contents

1 Introduction

There are many types of operational technology (OT) systems. The term is used to describe a broad range of systems that interact, and cause direct change with physical environment. An OT system can contain various types of control components, e.g., electrical, hydraulic, or mechanical. An OT system can be thought of consisting of two individual parts: process and control. Process is the part of the system that is responsible for producing an output while control (or controller) is concerned with maintaining conformance in regards to given specifications [41].

A controller can be seen as a small, real-time computer system that functions by controlling electrical outputs based on program logic. Different kind of field devices can be attached to a controller either directly or by a *fieldbus connection* [32].

Supervisory control and data acquisition (SCADA) systems are used to control and monitor various field devices [47]. These systems are commonly used in many distribution systems, such as water distribution, oil and natural gas pipelines, and rail and other public transportation systems [39, 41]. A SCADA system combines data acquisition, data transmission and Human-Machine interface (HMI) software into a single system providing centralized monitoring and control capabilities for numerous process inputs and outputs [7].

These SCADA systems collect information from field devices and transfer it to a control center where it is displayed to an operator graphically or textually in a central location [1]. These central locations are equipped with computation and communication facilities and often consist of data servers, HMI stations, data historians, and other servers that aid the operators in the overall management of the factory network [11, 47]. This type of real-time monitor and control system allows for any of its individual subsystems to be controlled either automatically, or via operator commands [15].

SCADA systems are considered to be hard real-time systems because completing an operation after its deadline has passed is considered to be useless [48]. The delay can also potentially cause cascading effects in the physical world. [49, 5]. If the system fails to react in a given time window this can have safety affecting effects such as damaging the surroundings or even threaten human lives.

The structure of this thesis is as follows: Chapter 2 gives an high level introduction to SCADA systems, from its beginnings to modern usage. Chapter 3 discusses security in SCADA systems on a general level. Following it, Chapter 4 offers some more concrete examples of hardening SCADA systems against various attack types. SCADA systems incorporate bits from both OT and IT systems. This notably broadens the landscape of available technologies and protocols which SCADA systems can potentially be build on. This thesis focuses its scope on communication protocol security in SCADA system. Basics of SCADA communication protocols are introduced in Chapter 5 and communication security in Chapter 6. Lastly, the main points of this thesis are discussed in Chapter 7.

2 SCADA

Contemporary SCADA systems are a magnitude more sophisticated than the first legacy SCADA systems deployed in the late 1960's over analog wiring. This chapter starts by looking at how the gap between OT and IT systems has decreased over time. After that the chapter discusses how industrial control systems evolved from analog technology into digital in the 1970s and how that enabled further development of SCADA systems. Lastly, the evolution of SCADA system architecture and its modern characteristics are examined.

2.1 History

SCADA systems have been used for almost 40 years [39]. These systems have become more advanced and complex over time and are now a crucial part for the uninterrupted operation of critical infrastructure[2]. In addition to monitoring and controlling processes and devices they also store data for auditing purposes.

Initially, OT systems differed substantially from information technology (IT) systems. OT systems were running on isolated networks and used proprietary control protocols on top of customized hardware and software [42]. The development of industrial communication networks picked up in the 1980s when new emerging technologies were incorporated from the information and communication technology (ICT) world [45].

The gap between OT and IT systems has become substantially narrower as low-cost Ethernet, Internet Protocol (IP) and wireless devices have become widely available for OT systems [2]. They are now replacing previously used, older, proprietary technologies [47]. As a side effect, this also exposes the OT systems to cybersecurity vulnerabilities that previously only affected IT-systems [41]. This increase in interconnectivity decreases the isolation of OT systems and thus increases the their exposed attack surface.

2.2 Modern usage

Contemporary OT has evolved as a consequence of IT capabilities being introduced into already existing physical systems by replacing or enhancing physical control mechanisms

[7]. This has been a key enabler in development of many modern smart technologies such as smart electric grid, smart buildings and the Internet of Things [41].

Modern industrial facilities are large, distributed complexes that require continuous monitoring and controlling of many different sections of the production facilities [7]. Enabling such remote command and control has required development of networking technologies; The first control networks were simple point-to-point networks between monitoring devices and remote sensors, whereas modern solutions are complex networks that support communication between a central control unit and multiple remote units on a common communication bus [15]. The nodes on these networks are specially embedded devices such as sensors, actuators and Programmable Logic Controllers (PLCs) [37].

2.3 Architecture overview

Contemporary SCADA systems are based on standard information technology utilities and protocols, such as, TCP/IP, other internet, and wireless technologies [47]. A key driver for this has been the need to increase interoperability and to drive costs down [1].

Time period	*Architectural style*	*Description*
1970's	Monolithic	Controlled units were on the same site as the controlling computer with hard-wired connections between them.
1980's-1990's	Distributed	SCADA systems networked with devices using special purpose protocols. No external network connection.
2000's	Networked	SCADA systems no longer isolated but connected to external networks, e.g., internet.
2010's	Web-based SCADA	Accessing SCADA components from every where at any time using any web browsers, thin clients, PDA, mobile phone, etc.
2020's	Agent-based SCADA	Using agents and multi-agent systems new architectural style to build scalable, reliable, and flexible agents.

Table 2.1: Chronological description of SCADA architecture evolution. Amended from [1].

Table 2.1 describes the evolution of SCADA's architectural style starting from the 1970s. The first SCADA systems were architecturally monolithic [47]. This means that both the controllers and the field devices were physically on the same site connected via hard-wired connections [2]. Next in the 1980s and 1990s SCADA architecture saw a shift into distributed solutions, devices were networked using proprietary, special purpose protocols but the system had yet no connection to external networks.

With the dawn of the millenia SCADA systems become networked. Meaning they were no longer isolated, but rather had connectivity to external networks, such as the Internet. With the advance of internet connectivity SCADA systems become increasingly web-based during the 2010's [42]. This made it possible to access SCADA system components remotely at will. The current predominant architectural for SCADA systems is agent-based SCADA. Sharing the same basic principle with microservices the modern SCADA systems use agents and multi-agent systems to achieve scalability, reliabity and flexibility.

Figure 2.1 depicts a general SCADA system layout. All SCADA systems contain a *control center* (grey background in the figure). It consists of a control server and communication routers. Connected to the control server are often the HMI, any engineering workstations used by operators, and a log keeping server (data historian) [47]. All devices within a control center are connected by local area network (LAN). A control center is responsible for collecting and logging information collected from field devices, relaying information to HMIs, and for generating actions based on information gathered from field devices.

Secondly, SCADA systems contain some *communication media* (blue background in Figure 2.1), such as WAN, satellite, radio, or a telephone line. Communication media and the appropriate hardware for it provide means to transfer information and data between the control center and field sites. This is done using both standard and proprietary communication protocols.

Lastly, SCADA systems have a varying number of *field sites* (orange background in Figure 2.1). Field sites contain remote terminal units (RTUs), PLCs, and intelligent electronic devices (IEDs) that can be accessed remotely. An example of an IED is an intelligent protective relay. Depending on the configuration, a local RTU may poll the IED and collect data or it can communicate directly with the control server. IEDs can also be used to expose a direct interface to control and monitor equipment and sensors on the field site. These devices and units are used to perform control actions and to monitor various sensors [34]. Operators in the control center are able to perform remote diagnostics and, if necessary, repairs in the field site using the remote access capabilities. This is usually

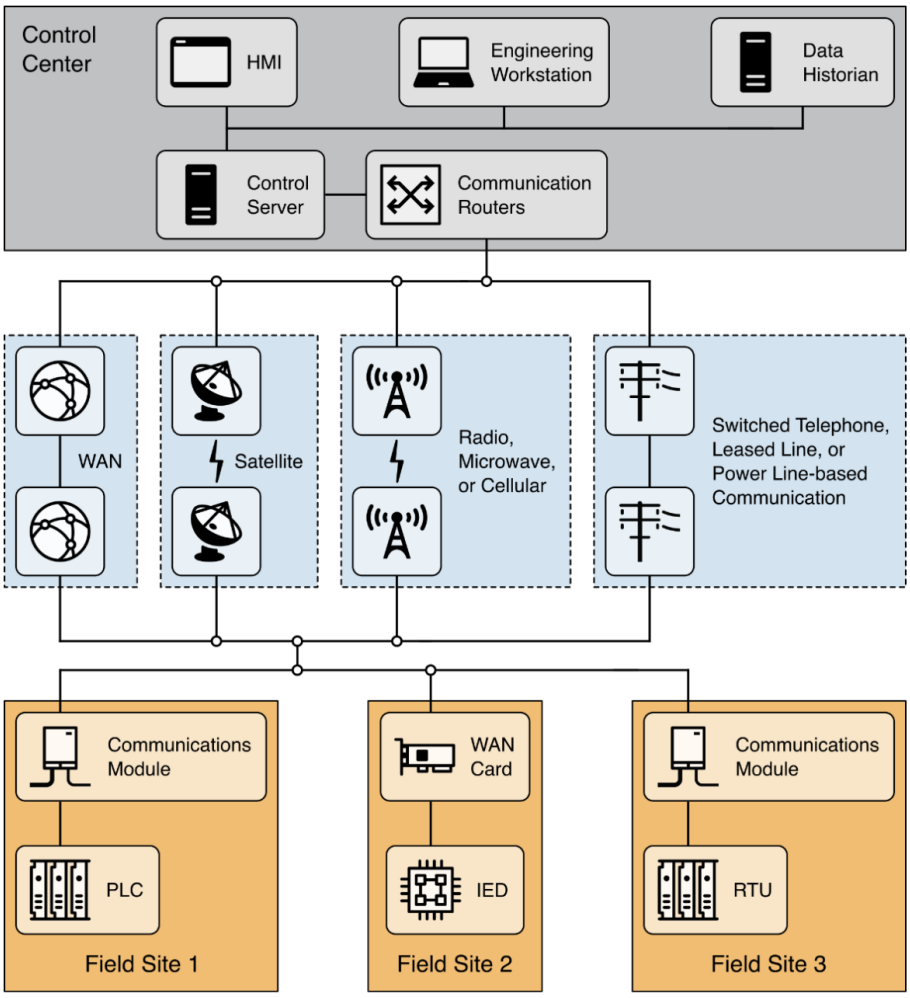

Figure 2.1: An example general SCADA system layout showing control center devices, communications equipment and field sites. Taken from [41].

done over a separate dial-up mode or WAN depending on the used connection type.

Authors in [42] divide the evolution of SCADA architecture model into four generations. This division is depicted in figure 2.2. The first generation model has an Master Terminal Unit (MTU) that functions as a mainframe system. The MTU serves as an HMI interface for the operators and is also responsible for communicating directly with PLCs and RTUs on field sites. The first generation architecture is a monolithic, closed system that utilizes only proprietary communication protocols.

The first generation systems used large minicomputers such as the PDP-11 series* for its computing needs [47]. The proprietary communication protocols used imposed some limitations due to nonexisting interoperatibility between different vendors. In practice, this

*http://www.pdp11.org/

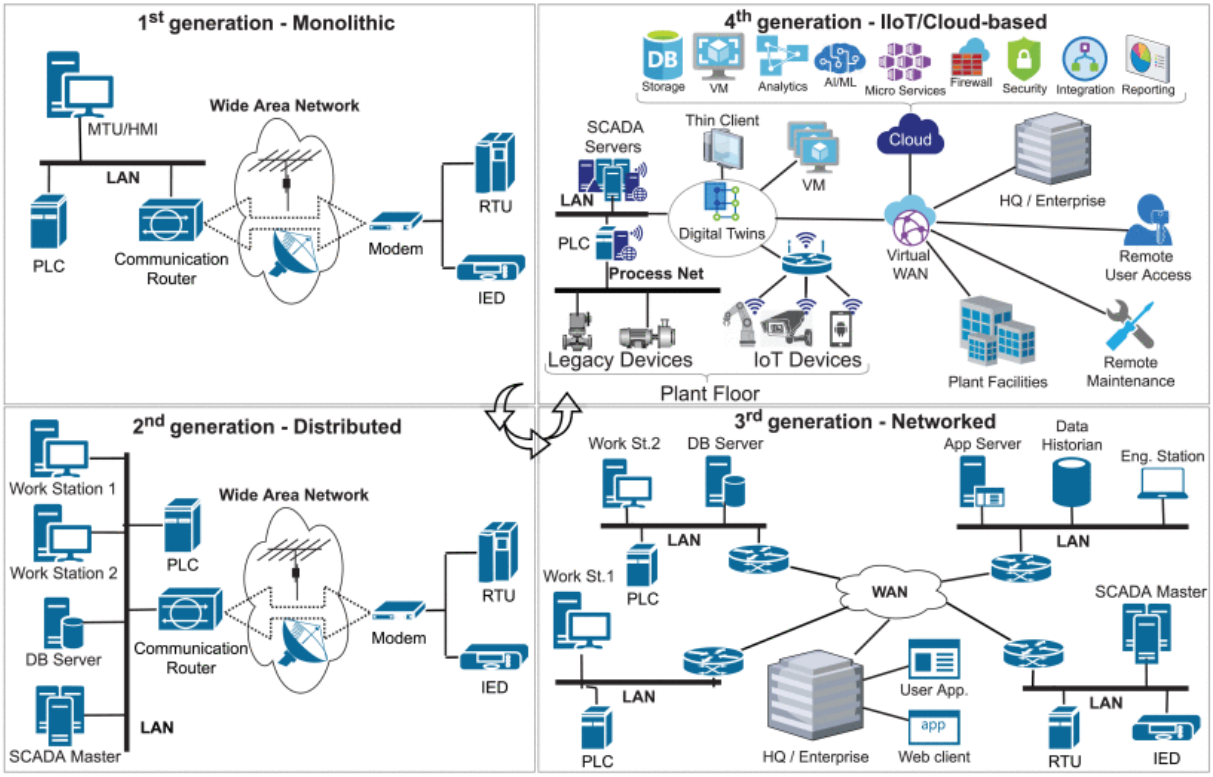

Figure 2.2: SCADA architecture evolution. Taken from [42].

meant that RTUs produced by different manufacturer could not be connected to another vendors MTU. Furthermore the protocols mainly supported only scanning, control and data exchange operations between the MTU and RTUs[47]. The connection between the MTU and RTUs was implemented at the bus level. One of the only ways to implement redundancy was to provide a equivalent, fully equipped system, connect it and place it on standby.

With **the second generation** architecture model SCADA systems switched from monolithic to a distributed model [42]. One of the fundamental changes was the introduction of redundancy. A control center has multiple servers and operator workstations. A separate mainframe functions as a SCADA master while logging is done on a dedicated data historian server. The increase in hardware also implies a noticeable increase in computing power.

These second generation SCADA systems were usually interconnected through Local Area Networks (LAN) [42]. The computation overhead was getting distributed between remotely located systems. The communication between field sites and the control center's MTU was often implemented over an Wide Area Network (WAN). The systems distributed

nature along with increased redundancy made them more reliable [47]. As was the case with first generation systems, the second generation systems also mainly used proprietary network protocols and components. This network isolation provided by proprietary protocols and software was the only tangible security hardening these systems had. Otherwise security was not really a concern.

The third generation architecture model leverages internet technologies. It allows for wider geographical dispersion of field sites as communication from and to various field sites takes place using standard TCP/IP procotol [42]. Adoption of internet protocols also make it possible to use off-the-shelf networking equipment and various open source solutions. This integration while making the model more open compared to previous generations also increases the available attack surface. Weaving standard IT networking protocols into the model also brings with it all the security vulnerabilities of those protocols [47].

These third generation SCADA systems are referred to as modern SCADA systems [42]. The big difference from second generation systems is the vast usage of open source protocols and standards. The usage of open standards makes it possible to interconnect third-party field devices to the network[47]. The introduction of internet protocols also brought with it a transformative improvement. These improved routing capabilities practically made the networks have, for the first time, disaster survivability.

The fourth generation model is heavily cloud-based [47]. It makes use of Industrial Internet of Things (IIoT) smart devices to increase data collection and process monitoring capabilities. Communication happens over Software Defined Networks (SDNs) where the network topology can be adjusted as needed and is not restricted by the physical layout. The architecture resembles a collection of subsystems where cloud services and virtualization create a service-oriented SCADA system.

Integration of Internet of Things (IoT) technologies and affordable cloud computing have significantly driven down the infrastructure and deployment costs of fourth generation SCADA systems [42]. Data exchange and field device control messages within the system are done, by and large, using open communication standards [47]. Big data analytics is leveraged to gather insights from the systems. These systems also employ data-driven techniques to identify anomalous behaviour.

Most of contemporary SCADA systems in use nowadays are either of 2nd or 3rd generation [42].

3 Security in SCADA systems

SCADA systems are used to supervise and control cyber-physical systems that interact with the physical world. These supervised cyber-physical systems can be e.g., waste-water disposal systems, smart electric grids or other critical infrastructure. Disturbances in their physical processes can lead to financial loss, loss of property and have cascading effects in other dependant systems or even cause risk to human safety. This chapter focuses on SCADA system security and discusses some quintessential security aspects of SCADA systems on a general level.

3.1 Overview

SCADA systems are designed to be fault-tolerant and to have significant redundancy built in [41]. SCADA systems differ from traditional IT systems in multiple aspects. One of the major differentiating factors is that any logic execution causes direct impact in the physical world. This has been used as an argument stating that from a safety standpoint, field devices are equally important in comparison to central hosts [49]. SCADA systems often have special requirements such as: demand for continuous availability, time-criticality and constrained computation resources on edge devices.

Due to physical nature, many of the tasks performed within a SCADA system need often to be interrupted and restarted. Time-criticality and task interrupts can hinder the usability of conventional encryption block algorithms. SCADA systems run certain operating systems and applications that are not off-the-shelf compatible with commercial IT cyber security solutions. This fact coupled with different performance and reliability requirements compared to IT-systems make securing a SCADA network its own, distinct challenge [41].

Many of the field devices in SCADA systems are embedded systems [13]. These can run years without a single reboot. Long system uptimes can imply that the devices accumulate fragmentation. This makes buffer overflow more problematic issue in SCADA than in traditional IT [49]. SCADA systems have many different characteristics from traditional IT systems, including different risks and priorities. Some of the risks include direct risk to the health and safety of human lives, serious damage to the environment, and financial

issues, such as production losses [12, 41].

3.2 Common security aspects and threats

SCADA systems are used to remotely manage station control devices or field devices: These systems provide real-time data about the monitored systems [8]. Industrial networks are run on remote, non-Internet connected locations [49]. This isolation provides security by removing the possibility to remotely connect to them. Industrial networks used to run proprietary and/or custom software and hardware stacks [2]. Nowadays, as a result of standardization efforts, industrial networks have adopted technologies and protocols such as Ethernet, TCP/IP, IEEE 802.x, and Bluetooth.

As a result of this trend SCADA systems are no longer protected by the obscurity of the used technology stack nor by systems being air-gapped [49]. As a consequence, planning and executing SCADA specific attacks has become easier. In the past, for one to be able to break into production networks they first had to find their way into an associated IT-system [12]. From the breached IT-system an attacker could then pivot laterally into Industrial Control Systems (ICSs) and from there infect the production environment.

Similarly as is the case with older internet protocols, SCADA protocols were initially developed with a goal of ensuring good performance and that any task constraints on the network would be met. Security was not woven into the protocol design as network security was not, nor was it seen, as a concern [15]. Increased connectivity between field sites and corporate networks have transformed SCADA networks from isolated, single user-based control networks into a part of a more complex interlinked systems with Internet connectivity [11].

SCADA networks, as a consequence of being connected to the company's corporate network and to the Internet, are exposing their often safety-critical industrial network to all the security problems of the Internet [15]. This means that attacks against a SCADA network might cause harm to the environment and even to public safety [12, 39].

Adoption of modern networking technologies brings with it the possibility of having multiple access points to any given network [42]. For SCADA networks this means that physical isolation should not be thought to ensure network security [47]. Instead, it should be taken into consideration that a malicious actor can attempt to take advantage of any connected communication medium within the SCADA network to gain access to either the control

center or to any of the field sites [15].

Traditionally, the trinity of desired security properties for a given IT system are confidentiality, integrity and availability, also known as the CIA triad [42, 49, 15]. Of these three, two most important properties are confidentiality and integrity as lack of availability does not eradicate trust on the data. However, in SCADA systems, the two most important properties are system availability and data integrity [15, 47]. This is due to the fact that human and plant safety is the system's primary concern and disruptions to either property can endanger this. Message integrity is critical in ensuring the correct operation and reliability of SCADA systems [5].

Many of the older, but still used, proprietary SCADA protocols do not natively support encryption [2, 5]. This makes it feasible for a suitably positioned malicious actor to be able to sniff network traffic within a SCADA system [8]. This could allow the attacker to listen and learn the data and control commands and later to use these, as such legitimate, commands to send false messages [15]. Another attack vector would allow an attacker to alter the operator display values in such a way that the HMI systems would falsely report acceptable values even when an alarm threshold has been reached.

Table 3.1 from [39] describes various threat and vulnerability concerns. It can be seen from the table that the bulk of the listed concerns are similar to what one would list for any arbitrary IT-system. In regards to network security, the threat impact is listed as *high* and the types of attack origins span from malicious actors (hackers) to operator errors.

The table lists various security weaknesses with different impact severity ratings ranging from *low* to *high*. A common nominator in all of these is failure to adhere to latest security practices either due to poor maintenance of software and hardware or due to poor personnel practices (poor software deveplopment and poor monitoring).

A common theme for system weaknesses and cybersecurity threats in Table 3.1 is unintended system access or malicious program execution. Out of these the *high* severity rating is given to direct and indirect cyberattacks and *low* severity to intended and unintended (via protocol vulnerabilities) connectivity issues. These threats can be seen arising from increased system complexity. As complexity increases, so does the amount of loopholes and possibilities for misconfigurations.

A significant attack vector for SCADA systems emerges from the geographical distribution of field sites [47]. Physical security is a concern with all remote sites. Additionally devices on field sites are expected to run without interruptions. This makes installing software

Threats/vulnerabilities	Types	Description	Impact
Network security	Hackers	Unauthorized access to SCADA components Causing interruption of critical services	High
	Malware	Attacking SCADA applications used to monitor and manage SCADA systems	High
	Inside Error	Causing damage to SCADA systems as a result of operator errors	High
Security weaknesses	Poor staff training	Lacking training in the prevention, monitoring, and identification of potential security breaches	High
	Application development loopholes	Failing to enforce sufficient security requirements for critical SCADA system components.	High
	Maintenance issues	Unpatches and/or outdated software	Medium
	Connectivity	Accrued technical dept by leaving old field devices connected to the system	Low
System weaknesses	Commodity software and hardware	Inherit any existing vulnerabilities used protocols or software e.g., Microsoft, TCP/IP etc.	Low
	PC controllers	Novel microprocessors and embedded OSs bringing forth new challenges due to increased system complexity	Medium
	Connectivity to external sources	Control systems are controlled by remote access for troubleshooting purposes by third parties.	Low
Cybersecurity	Direct cyberattacks	Unauthorized access to computers and networks via network infrastructure	High
	Collateral cyberattacks	Worms, viruses or system failures caused by malware of staff member	High

Table 3.1: Threats and vulnerabilites against a SCADA system. Amended from [39].

patches and upgrades a more complex task. The devices are expected to not have any downtime as downtime means interruptions for production.

3.3 ICS system attack characteristics

Attacks against Industrial Control Systems (ICS) have certain differences compared to attacks targeting IT systems. An ICS system can be visualized as three distinct layers [2]. At the bottom is a physical layer that manages the controlled physical processes by utilizing various sensors and actuators. On top of the physical layer lays the ICS layer wherein SCADA systems reside. All process monitoring happens in this middle layer. The third, topmost, layer is the IT layer that houses all networks and systems build with IT technologies. Some of these systems are Internet-facing, e.g., corporate networks.

Figure 3.1 depicts these three layers and and describes them as stages through which an adversary has to breach in order to successfully execute an attack. The IT layer can be breached for example by leveraging social engineering tactics to deliver malware on to workstations in the company's network [2]. This initial foothold is then used as a staging point to launch further attacks against the ICS layer.

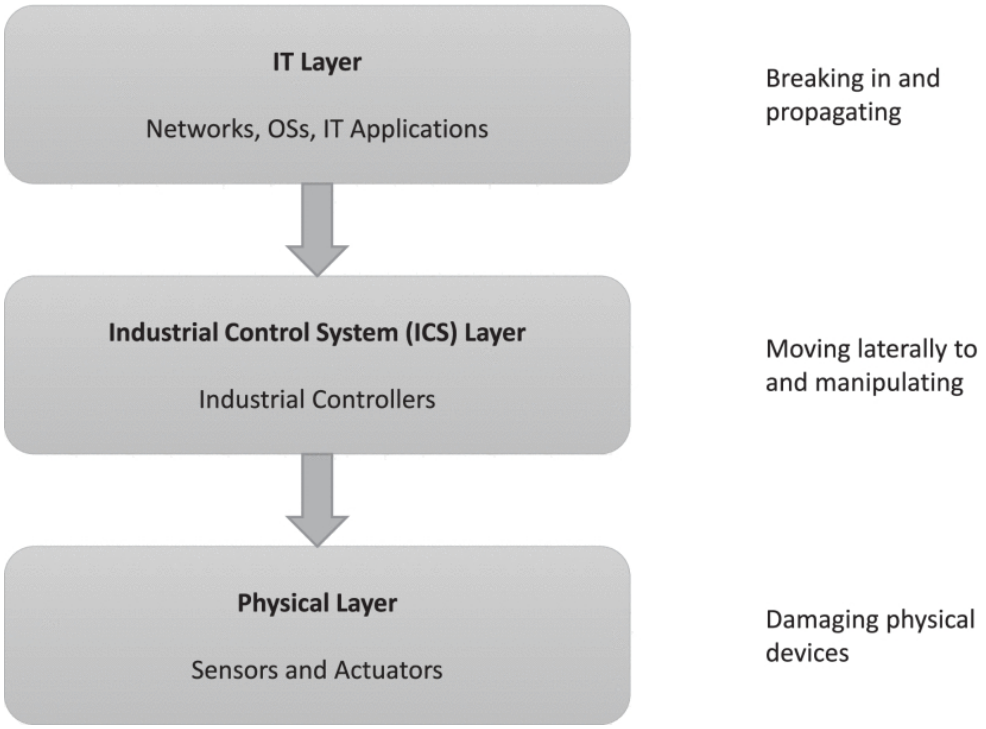

Figure 3.1: Tree stages of a typical industrial control system attack [2].

Many contemporary SCADA systems are integrated with IT networks [35]. This integration offers Internet connected remote access capabilities that an adversary can also leverage. The communication medium used for data exchange between the field sites and the control center is often accessible over Internet [33]. A malicious actor can leverage any of these connection points to compromise a field device and thus gain access into the ICS layer. After gaining access to the ICS layer, they are able to access field devices and manipulate them to e.g., cause disruption or even physical damage to devices and the controlled processes.

Whereas an attack against a traditional IT system requires breaching the IT layer, an attack against an ICS system requires the attacker to breach both the IT and the ICS layers [2]. Moving inside the ICS layer does not drastically differ, in principle, from the lateral movement performed in more traditional attacks that take place solely inside the IT layer [34]. In both cases the malicious actors move from host to host by leveraging some system component or software vulnerability to infect devices in order to gain foothold and to establish persistence in the system.

3.4 Known ICS cyber incidents

Modern literature knows an increasing amount of documented cyber incidents against targets employing SCADA systems. Figure 3.2 shows some instances of notable incidents and their impacts over the years. It can however be assumed that many more incidents remain still undisclosed and unknown to the wider public [47].

Below are listed some of the most widely reported ICS cyber incidents, some of which, but not all, are also included in Figure 3.2.

The first known cyber attack involving a SCADA system, **the Siberian Gas Pipeline Explosion**, occurred in 1982 [47]. The incident took place in Russia where a trojan was used to cause and explosion in a natural gas pipeline.

The **Stuxnet** worm targeted Iranian industrial control systems in 2011. It is suspected to have been created by the US and Israeli governments [34]. Stuxnet demonstrated that cyber attacks targeting cyberphysical systems can cause physical harm and damage equipment [35]. It was a wake-up call for organizations around the world to invest and improve in their cyber security practices.

The **BlackEnergy** attack caused widespread power outages in Ukraine in December 2015.

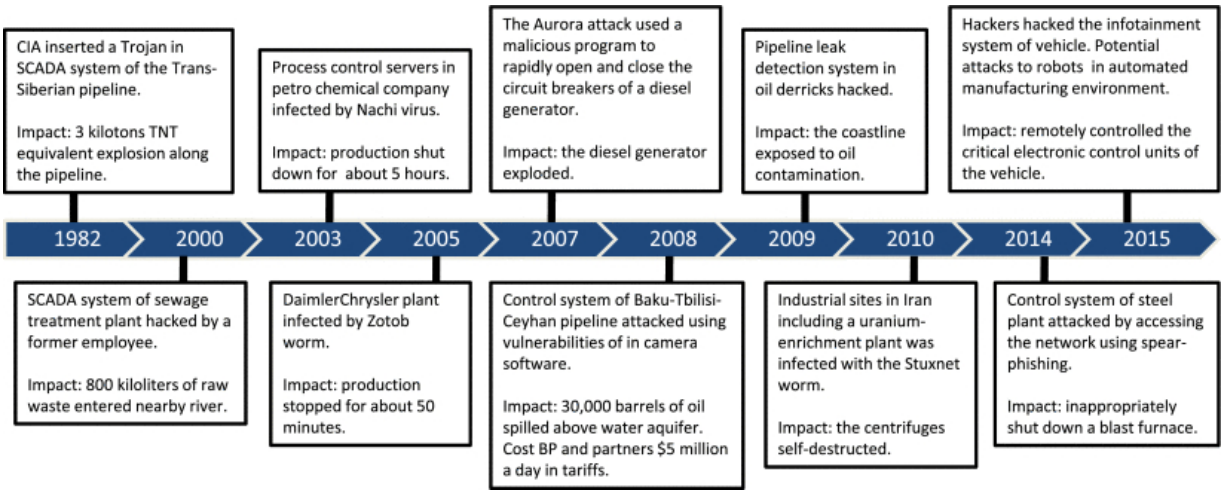

Figure 3.2: Timeline of cyber attacks against ICS systems and their physical impacts [33].

It is the first known cyberattack that caused widespread disruption to a power grid [34].

Triton was discovered in 2017. It was designed to specifically target ICS systems. This malware is considered to be highly sophisticated and it is capable of evading many traditional detection measures [34].

4 Mitigating against common attacks

Literature presents a myriad of different strategies and approaches to harden SCADA systems. Considerable portion of these approaches are better suited for contemporary SCADA systems that are not burdened with system requirements for long-term backwards compatibility and legacy devices with their limited computational capabilities. This chapter examines concrete defense strategy options, first some non-intrusive ones for legacy SCADA systems, and then a novel option for contemporary SCADA systems.

4.1 Protection for legacy SCADA systems

Legacy SCADA systems often utilize proprietary, legacy industrial communication protocols. These protocols send and receive messages in plain-text while having no built-in message authentication or encryption capabilities [35]. SCADA systems are used to monitor and control field sites that are interconnected and distributed over large geographically areas away from the control center. These field sites house cyber-physical field devices that are deployed to control and monitor critical infrastructure such as public transportation systems or electric power grids [44].

Ensuring the integrity of the data transferred from and to the field sites is paramount for correct functioning of the infrastructure managed by a SCADA system to function correctly [5]. Disruptions in the operation of the monitored physical process can have cascading effects. Disturbances in, e.g., smart power grids may collaterally affect other critical infrastructure that operate on electricity [4, 6]. Yet most SCADA systems deployed before year 2000 run legacy ICS protocols that were not designed to offer means to ensure message integrity. Instead, the design was heavily focused on reliability [35].

The rationale for leaving security considerations in the side rails was justified by the operation environment of older ICS systems. At the time of their devising these systems were running solely on isolated networks with obscure, proprietary protocols [4]. This is no longer the case and today's threat landscape contains a new set of security threats. There have been cases of physical break-ins to remote field sites [6]. Additionally, there

are documented cases where computer worms have been found that specifically target ICS platforms via infecting field devices, specifically PLCs [37].

Increasing need to integrate legacy SCADA systems with more open systems such as corporate networks leads to increased attack surface [46]. This makes field devices in a ICS systems more open to malicious actors. A suitably positioned attacker with access to corporate network attached to a SCADA system could forge data packets or send forged commands. This is known as *false command* (*FCA*) or *false data injection* (*FDI*) attack [5]. Forged sensor values received by the control system and altered commands received and executed by actuators have potential to cause physical damage and malfunctions in the underlying physical system [4].

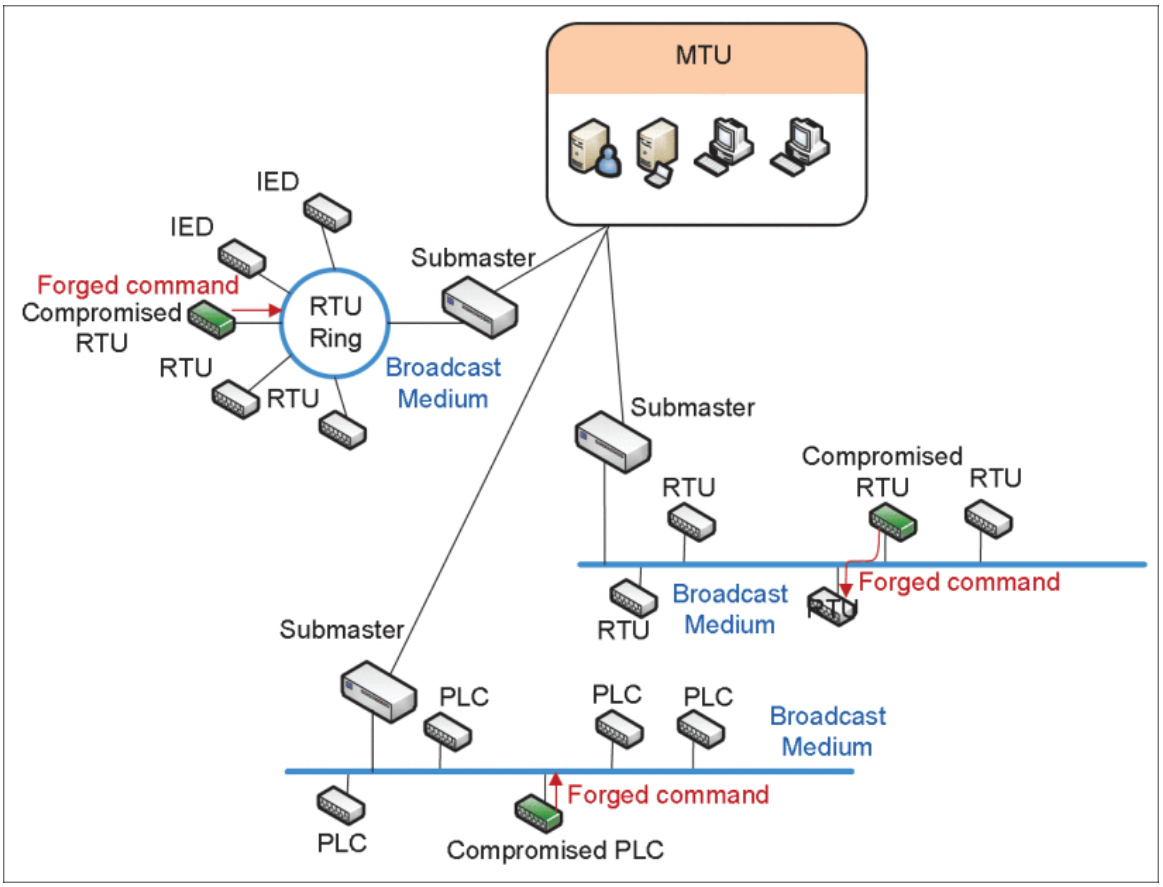

Figure 4.1: A typical legacy SCADA system and the possibility of false command attacks [6].

Figure 4.1 depicts an example legacy SCADA system and possible origin points (displayed with red arrows) for attackers to initiate FCA and FDI attacks. Fields sites in a SCADA system house a varying amount of PLCs, RTUs, and Intelligent Electronic Devices (IEDs). Each field device has sensors and/or actuators that are used to monitor and control parts of the physical processes [6]. Neighboring field devices in the same field site are intercon-

nected through a local field network that is also connected over a broadcast medium to a data concentrator, or a submaster. Each submaster, in turn, is connected to the Master Terminal Unit (MTU) in the control center's control server. Field sites may employ varying network topologies to form their respective field network. The broadcast medium (depicted with blue lines) which connects the field network to a local data concentrator is in legacy systems often implemented using either *EIA-485* or *DNP3* protocols [5].

These broadcast mediums open up an attack vector for forged commands to be injected in to the system. This can be done, .e.g, by using a compromised RTU or their own attacking device to masquerade as the MTU and issue false commands to other field devices. This has been demonstrated to be feasible in some legacy systems by using only off-the-self protocol analyzer software to eavesdrop control messages and to craft false command messages accordingly [6]. All legacy industrial communication protocols that do not offer authentication functionality are vulnerable to this type of attack [4].

False data injection attacks could be effectively mitigated by implementing a message encryption solution. Deploying such a solution to legacy SCADA systems is, however, often difficult [6]. Legacy SCADA systems are equipped with older-generation field devices that have limited resources. They thus lack the computational power to handle the processing overhead introduced by encryption. Additionally, introducing data authentication functionality to legacy, proprietary, SCADA protocols causes non-trivial compatibility issues [5].

Table 4.1 lists four different non-intrusive defense mechanisms and some selected key characteristics for each. For each mechanism the table lists whether either protocol or device level modifications are required, does the method support two-way communication from

	Bump-in-the-wire	**Data diode**	**Protocol-compliant authentication**	**Detect and respond approach**
Protocol-level modification required	X	-	-	-
Device-level modification required	-	-	X	-
Two-way communications supported	X	-	X	X
Security assurance	High	Very High	High	High
Level of customization required	Low	Low	Medium to High	High
Number of add-on devices required	One per field device	One per field network	Zero	One per field network

Table 4.1: Comparison of non-intrusive defense mechanisms for legacy SCADA systems. Amended from [6].

and to the legacy SCADA system to adjacent networks, the number of required add-on devices as well as a rough estimate on the level of customization required for implementation. Additionally, the table also lists the level of security assurance the mechanisms offer.

Bump-in-the-wire is a method in which all communication between the MTU and field devices is encrypted using a separate cryptographic device [46]. This approach requires that all networking equipment in the network that reside between the MTU and each field device are modified. The modification is needed to support a new protocol message structure that includes authentication fields for authentication data [4]. While the bump-in-the-wire method works with unmodified field device software, an additional device accompanying each field device makes this method inefficient for larger systems.

Data diode is a unidirectional gateway that can be used to enforce network restrictions. The unidirectionality is enforced by leveraging a physical attribute of a communication medium, e.g., by using a unidirectional optical coupler. Alternatively, with newer systems, this same one-way principle can be achieved by using Software Defined Networking (SDN) approaches [9]. A data diode solution can be used to provide a secure medium to transfer data from a more secure network to a less secure one. However, most data transfer mediums in SCADA systems are required to provide two-way data flow for the system to function correctly [5]. This greatly limits the applicability of this solution.

Protocol-compliant authentication is an approach where authentication data is embedded in the payload fields of the used SCADA communication protocols [4]. As the used protocol's packet structure remains unmodified, this approach does not require modifying the used networking equipment. However, modifications are needed for the software running on field devices as the authentication data has to be placed and extracted from the legacy communication protocol's payload [5].

Detect and respond approach is similar to the bump-in-the-wire method in that an add-on device called protection agent is installed in each field network. The number of additional installed devices is, however, significantly smaller as there is no requirement to install an add-on device (in this approach, a protection agent) for every single field device [6]. The protection agents' role is to act as a trusted relay from and to the MTU. They communicate with the MTU via an authenticated channel that is separate from the SCADA systems network. Thus, this approach does not require modifications in used networking equipment, SCADA protocols or field devices.

4.2 SCAPHY attack detection method

As an example of an mitigation method against some of the security threats discussed in Section 3.2, we will now take a closer look SCAPHY, an attack detection method developed for SCADA systems.

Ike et al. [16] present a new hybrid technique to detect industrial control system (ICS) attacks by correlating SCADA execution phase-specific behaviours with their impact in the physical world. It is built around the idea of detecting abnormal activities in a SCADA system by building a physical model of the system. The model is used to determine the set of legitimate behaviours of the system. Knowing the systems behaviour in each execution phase allows for the detection of any attack behaviours that violate process-control phases.

Industrial control system attacks can be divided into two categories based on the attacks inflicted behaviour: noisy and not noisy. Examples of the former are network scans and malformed protocols whereas the latter only relies on valid protocols and actions.

Ike et al. [16] argue that modern ICS attacks are able to evade statistical analysis of ICS traffic due to the fact that they leverage legitimate protocols and valid ICS parameters to cause disruptions. A further point is raised that prior ICS tools work by analyzing network traffic and sensor data in isolation and thus lack the ability to tie the produced analysis results into attack-execution context.

In practice this means that using merely statistical analysis tools in an attempt to identify malicious activity in a SCADA system is likely to yield poor results. The reason for this is that modern attacks attempt to cause disruption by performing perfectly legitimate actions, albeit at unconventional phases. Thus they will not pop up as outliers when compared against the default set of execution commands.

Ike et al. [16] explain that SCAPHY, their hybrid technique, generates a physical model of a given SCADA system by using open platform communications (OPC) standard conventions. It is able to map SCADA elements to their corresponding processes on the basis of novel process dependency and impact graph (PDIG) model. Additionally, SCAPHY is capable of identifying legitimate process-control behaviours.

SCAPHY uses a SCADA engine, a simulator tool, to perform a *physical process-aware* dynamical analysis during which legitimate process behaviour of the inspected SCADA system is learned. PHYSical world Impact Call Specialization (PHYSICS) constraints are a set of constraints that SCAPHY establish and then leverages to distinguish legitimate

behaviours from anomalious ones. Once the PHYSICS constrains are known, malicious activity can be identified by correlating observed behaviours in SCADA and in physical world. SCAPHY is said to use its physical mode to detect when control signals cause a physical process to have inconsistent state or is driven outside its setpoint ranges.

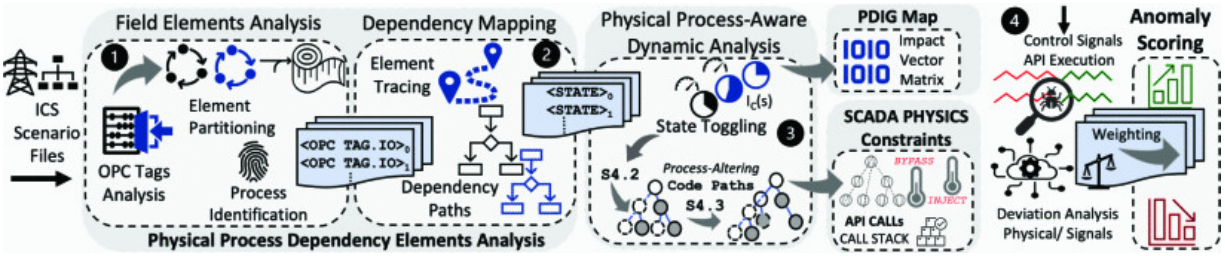

Figure 4.2: Illustration of the four phases of SCAPHY architecture. Taken from [16].

Figure 4.2 depicts SCAPHY's operating principle as presented in [16]. The authors explain the execution of each phase as follows:

1. At this phase SCAPHY analyzes ICS scenario files in an attempt to detect and partition ICS elements into two distinct sets: terminal and non-terminal. This is done by applying heuristics to OPC conventions.

2. Connections of each identified ICS element are traced in order to map the dependent element onto their corresponding process. Next, the created scenario is loaded in an ICS engine and a physical process-aware dynamic analysis is performed.

3. The ICS engine execute various code paths of process-control operations by iteratively switching the mapped ICS element states. As this is taking place, all executed API calls are recorded. These calls are then used to learn the PHYSICS constraints.

4. Whenever in any process-control phase an executed API is not in the PHYSICS constraints SCAPHY raises alarms.

Ike et al. [16] discuss SCAPHYs limitations and state that it is unable to detect any such attack that originates from outside the SCADA system. Examples of such attack origins are side channel attacks and device hardware originating attacks. The authors conclude by noting that in their testing, SCAPHY outperformed existing detection tools and achieved high accuracy rate.

5 Communication protocols and SCADA

As SCADA systems consist of geographically scattered field sites and central command centers, communication protocols used in these systems play a vital role. They have to ensure that all communication and transmitted data retain their integrity and meet the systems' real-time requirements. Failure to adhere to these requirements can have far-reaching consequences to their physical processes. This chapter focuses on SCADA communication protocols; their history and evolution, and key characteristics.

5.1 Overview

Communication protocols are regulations that define how data should be depicted and how it can be exchanged over a communication link [47]. Communication protocols enable data exchange between a control center and field sites but also more broadly between all devices connected to a SCADA network [14]. In older SCADA systems devices were connected using *RS-232*[†] (*Recommend Standard 232*) protocol, more commonly known as *serial ports*.

Early Industrial Control Systems (ICS) starting from the 1970's were built for each use case separately [47]. Communication was done over fixed wiring and the network often had custom designs tailored specifically for the particular enterprise's control requirements [2]. Control applications used for monitoring were custom made and the data from and to the field sites ran in physically isolated networks.

The added level of security provided by the physical isolation was one reason why many earlier industrial communication protocol versions, such as EtherCAT[‡] or Modbus[§] did not support authentication or encryption [2].

Industrial communication in early ICSs took place over parallel cabling. Controllers and field devices were directly interconnected. Dedicated automation networks, *fieldbus sys-*

[†] https://www.telecomabc.com/r/rs232.html
[‡] https://www.ethercat.org/default.htm
[§] http://www.modbus.org/docs/PI_MBUS_300.pdf

tems were developed as a solution to solve the limitations of starlike point-to-point connections done over serial cabling [45]. The term *fieldbus system* originates from the process field of a production factory [40]. Fieldbus networks allowed for modular installations where field devices could be added and removed from the network with more ease.

Field-level networks were originally built with different system objectives than those of computer networks, especially regarding data and traffic quality [40]. Generic *local area network (LAN)* characteristics include high data rates and big data packet sizes whereas field-level networks transport mainly process data, hence the data rates are lower and packet sizes are smaller. This, in part, allows meeting the crucial real-time requirement of ICS communication.

Internet technology became increasingly popular around the change of the millennium. Simultaneously with this, a wave of Ethernet-based networks were introduced to industrial automation. These network technologies incorporated basic solutions and protocols from the IT world [45]. The boundary between automation and IT networks has become somewhat blurry in recent years. It can be argued that nowadays the way to define a field-level network is via application viewpoint and not through whether or not the network is used for automation [40].

A security consideration for all automation networks is that field network installations have considerably longer lifetimes compared to IT systems. A typical lifespan for a IT network can be estimated to be roughly around three years whereas for automation networks it is more than ten years [40]. This difference also comes in to play regarding field-level communication protocol development. New implementations need to provide backward compatibility support for long periods.

A SCADA system can be viewed as a large collection of various, often proprietary, components [7]. Many component manufacturers implement their own, vendor-specific protocols that cannot be used to communicate with other components that use some other manufacturer's protocol. The need to improve the interoperability of SCADA system components and to scale the systems up while pushing costs down has been a driver to create open source protocols [31].

Implementing open protocols to SCADA systems increases the system's vendor independence and reduces cost while allowing for easier technical support. *Distributed Network Protocol 3* (DNP3) is an example of an open source SCADA protocol [14]. A main motive for its development was to provide an open and standards-based communication protocol for SCADA systems [47]. The original DNP3 protocol only had support for slow serial

interfaces. Newer versions, however, do support TCP/IP [38].

SCADA system are comprised of components [38]. The main components, that on a high level allow the system to function are the *Operator*, the *HMI*, the *Intranet*, the *Control Server*, the *Remote Terminal Unit (RTU)*, and *Field Devices*. Operators can be located either on-premise at the facility or they may access the system remotely [47]. The operators view the system state through an HMI and issue control commands as needed.

The HMI gathers information from the Control Server. The Control Server resides in a control center and is responsible for gathering data from field sites and passing it on to HMI, as well as to a logging solution. The Control Server manages the high-level control logic for the system and is responsible for forwarding control signals to the field sites.

Remote Terminal Units reside on the field sites and are responsible for the communication from and to the Control Server [12]. They thus act as communication gateways for their respective fieldsites. The RTUs also manage relaying control signals to field devices [5]. Field Devices are devices that monitor and control the physical process. An individual device can be, e.g., a sensor that gathers data or an actuator that performs controlling actions. Contemporary, larger SCADA systems can contain several hundreds of thousands of field devices that send and store similar number of messages each second [7].

All components need to be able to communicate between themselves. This communication needs to happen continuously, reliably and efficiently. To meet this need, a number of communication protocols have been created [45]. Protocols intended for Industrial Communication System (ICS) setting need to take into consideration the partly limited processing capabilities of the components. A key limitation comes from the various field devices that are often embedded devices with limited resources [6]. Another key consideration for the protocols is about the communication requirements of the industrial applications, e.g., data integrity is paramount.

5.2 Protocols

Table 5.1 shows a list of various SCADA protocols, as presented in a 2020 literature survey [38]. This list is not an exhaustive compilation of all communication protocols used in SCADA systems, rather it is a summary of protocols encountered during the survey. The table shows protocols in alphabetic order. The following attributes are listed for each protocol: supported network infrastructure types (transmission medium) , supported

Protocol	Network Infrastructure	Topologies	Data Rates	Maximum Distance
BITBUS	Fieldbus	Bus	62.5 Kbps, 375 Kbps, 1.5 Mbps	1200 m
DC-BUS	2-wire cable	Line	115.2 Kbps up to 1.3 Mbps	100 km
Distributed Network Protocol 3 (DNP 3)	Ethernet	Line, Peer-to-Peer	100 Mbps, 1 Gbps	100 m
EtherCAT	Ethernet	Ring, Line, Daisy-chain	100 Mbps	100 m
Ethernet Powerlink	Ethernet	Tree, Line, Star, Peer-to-Peer	100 Mbps	100 m
Foundation Fieldbus H1	Fieldbus	Point-to-point, Bus with Spur, Daisy-chain, Tree	31.25 Kbps	1900 m without repeater, 9500 m with up to 4 repeaters
Foundation HSE	Ethernet	Tree, Line, Star, Peer-to-Peer	100 Mbps	100 m
HART	2-wire cable	Point-to-point, Multi-drop	1.2 Kbps	3 km
IEC 60870	Serial, Ethernet	Ring, Tree, Line, Star	N/A	N/A
IEC 61850	Ethernet	Ring, Tree, Line, Star	N/A	100 m
Modbus	Serial, Ethernet	Line, Star, Ring, Mesh (with MB+)	100 Mbps, 1 Gbps	N/A
PROFIBUS	Fieldbus	Point-to-point, Bus with spur, Daisy-chain, Tree	9.6 Kbps to 12 Mbps	100 m to 1200 m, 15 km for optical channel
PROFINET	Ethernet	Ring, Tree, Line, Star	100 Mbps, 1 Gbps	100 m
RAPIEnet	Ethernet	Line, Ring	100 Mbps	100 m
SERCOS III	Ethernet	Line, Ring	100 Mbps, 1 Gbps	N/A
Unitronics PCOM	Serial, Ethernet	Ring, Line, Star	100 Mbps	100 m
WorldFIP	Fieldbus	Bus	31.25 Kbps, 1 Mbps, 2.5 Mbps, 5 Mbps	1 km

Table 5.1: SCADA communication protocols [38].

network topologies, achieved data transfer rates, and maximum cable length.

Next we discuss examples of some of these communication protocols.

Fieldbus-based protocols

Pliatsios et al. [38] summarize the nature of fieldbus-based protocols: Fieldbus is a network system that can be used for controlling and monitoring real-time industrial systems. It is a way to connect various field devices to their corresponding controllers. Different fieldbuses vary in terms of what types of network topologies, transmission mediums, and transmission protocols they support.

Fieldbus provides shorter path lengths between the nodes on a SCADA network when compared to parallel wiring. A fieldbus is a single cable to which each device in the network is directly connected to. By having shorter paths the network's reliability and thus availability increases. This single cable-structure makes designing and deploying networks easier. It also allows for the network to be modified in the future according to changing requirements.

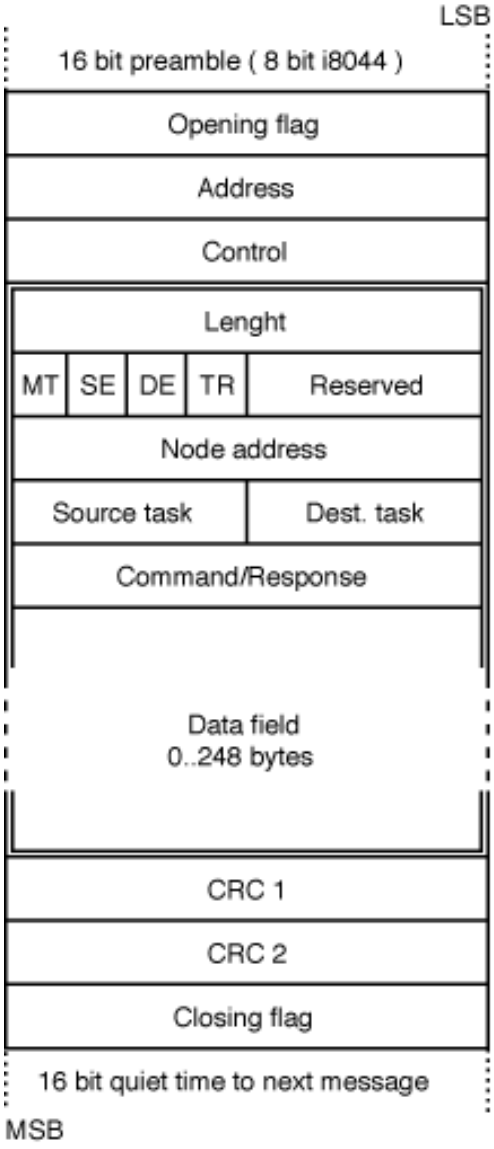

Figure 5.1: A BITBUS frame structure [3].

BITBUS [3] is an open, non-proprietary fieldbus-based protocol. It is also known as IEEE-1118. It supports up to 1200 m cable length and up to 1.5 Mbps transfer rate depending on the cable distance. BITBUS supports connecting up to 250 nodes in single bus if repeaters are used. This, however, reduces the available data transfer rate.

Figure 5.1 shows BITBUS message structure. A BITBUS message is encapsulated in a *Synchronous Data Link Control* (SDLC) frame. This SDLC frame wraps the BITBUS message as its payload. Each BITBUS frame starts with a preamble that is followed by Opening flag field. The following fields are common packet headers seen in most transmission protocol specifications: Address field, four Control fields for routing information

(MT, SE, DE, TR), Length, Control flags, and Node address. The Source and Destination tasks are used to identify the task from which the message originates from. The Command/Response field identifies the command to be executed or if an error has occurred, an error code. The CRC error codes and the ending flag are not part of the BITBUS frame but belong to the SDLC frame.

Foundation Fieldbus H1 protocol [43] can be used for communication between the control system and field devices as well as from field device to field device. The protocol is based on the ISO/OSI communication model, however, it does not include all layers. The foundation fieldbus protocol stack is divided into physical layer and communication stack. This maps to the OSI 7-layer model such that the physical layer is OSI layer 1 and the Communication stack encompasses OSI layers 2-7. The protocol includes very little cross-network communication and as such it does not contain layers that would map to OSI layers 3-6.

Foundation H1 has a fixed data rate of 31.25 Kbps and supports a maximum network length of 1900 meters. However, up to 4 repeaters can be used to extend the maximum distance between any two devices in the network up to 9500 meters. Maximum number of nodes in the network is 255 and each device is assigned an individual address. There is no support for broadcasting functionalities.

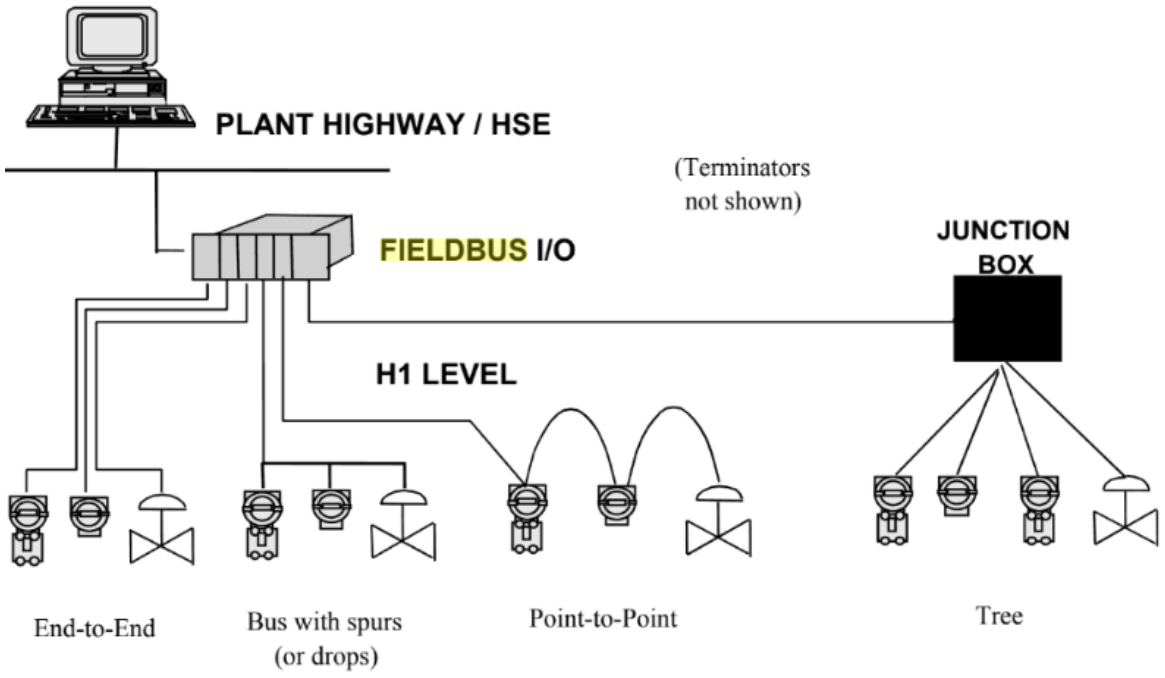

Figure 5.2: Foundation Fieldbus H1 topologies [43].

Figure 5.2 shows different supported network topologies. These are End-to-End, Bus with spurs, Point-to-Point, and Tree. End-to-End topology is used to directly connect two devices. The connection may take place entirely in a field site connecting two field devices or a field device may be connected to a linking device which in turn is connected to the control center. Bus with spurs topology connects devices and spurs directly to a single bus. Several devices may be in turn connected to a given spur. In a Point-to-Point (or Daisy Chain) topology, all field devices are connected in series. The bus trunk gets routed by being interconnected to each network device. A Tree topology uses a junction box to concentrate several field device connections.

Ethernet-based protocols

Incorporating Ethernet into industrial communication protocols allows for the protocols to leverage the advantages of it. Compared to prior legacy serial communication solutions, Ethernet allows for orders of magnitude greater transfer speeds. It additionally allows protocols to leverage Ethernet's error detection and error correction capabilities [38]. The original Ethernet standard, however, lacked genuine real-time capabilities and this led to the development of multiple Ethernet-based solutions for automation networks [45]. Industrial Ethernet protocols of today are results of this development. Today's various Ethernet-based industrial communication protocols only share commonalities in the data-link layer. Each implementation has it specific intrigues above the data-link layer [40].

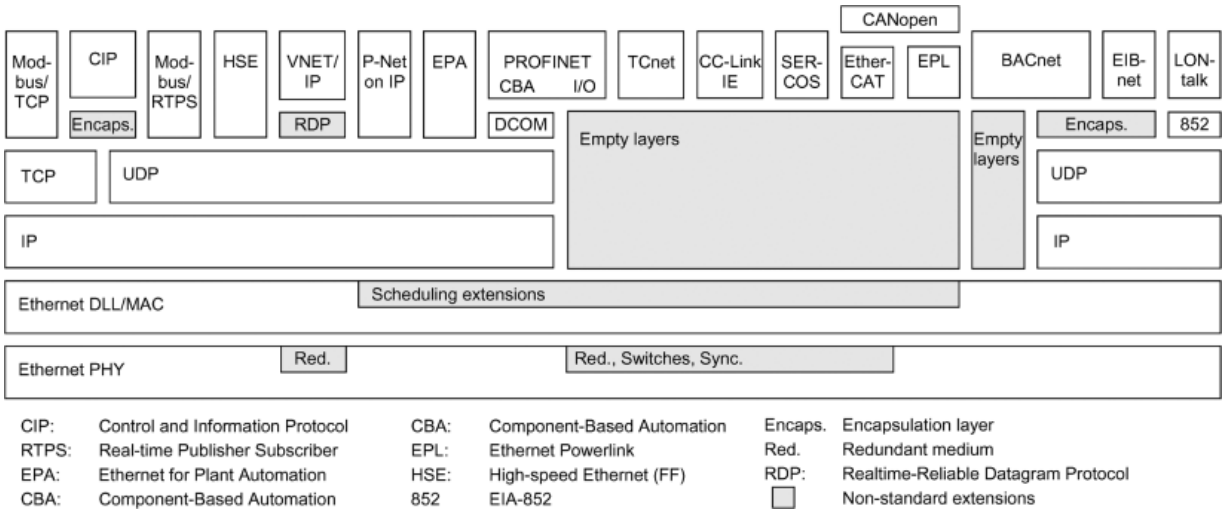

Figure 5.3: Protocol architecture of selected real-time Ethernet solutions for Ethernet-based building automation networks [40].

Figure 5.3 from [40] depicts the protocol stacks of some Etherner-based ICS protocols.

The figure is read column-wise from top to bottom. The top row lists a number of ICS protocols. Each row below an ICS protocol shows a protocol used by that ICS protocol, going from higher abstraction level to lower, i.e., from application layer all the way down to the physical layer. For example, the *Modbus/TCP* protocol utilizes the TCP/IP stack and Ethernet for data-link and physical layers (Ethernet DLL and Ethernet PHY). Some of the ICS protocols do not make use of each abstraction layer, e.g., CC-Link IE which uses existing CC-Link protocol over Ethernet and does not thus make use of TCP/UDP and IP protocols. Some protocols such as EtherCAT and EPL use also the CANopen in their application layer implementation to offer compatibility with some commonplace device profiles.

As Figure 5.3 illustrates, there are multiple variations of how Ethernet-based protocols are designed. Some are entirely new implementations with no backward compatibility in regards to older fieldbus solutions [45]. However, those Ethernet protocols that were developed by the same companies that themselves were offering fieldbus-based solutions did implement backward compatibility options that allowed the interconnection between their old fieldbuses and the Ethernet solution [40].

Distributed Network Protocol 3 (**DNP3**) is an open source, distributed network protocol used in SCADA systems [36]. It is based on the *Enhanced Performance Architecture* (*EPA*) model which is a trimmed down type of architecture of the OSI layer architecture [47]. The protocol defines three layers: *application layer*, *data link layer* and *physical layer* [8]. This protocol stack is illustrated in Figure 5.4. There is also a pseudo-transport layer that bi-directionally forwards segmented data units from the application layer to the data link layer. The application layer offers standardized functions and formats that are aimed to provide efficient data and control command transmission [14]. Data transmitted and received is organized into fragments by the application layer. Figure 5.5 shows the DNP3 request and response fragment structures.

Both fragment types have similar, but slightly different structures. The response header contains an additional field called *internal indications* which is not present in an application request header [14]. The headers begin with a control header that contains information on how fragment should be processed upon receiving. The fragment is then followed by zero or more data object headers. A data object header alone contains often complete information and no associated data objects are needed. The headers are followed by a number of DNP3 objects and their corresponding headers.

The data link layer functions by providing an interface that sits between the physical

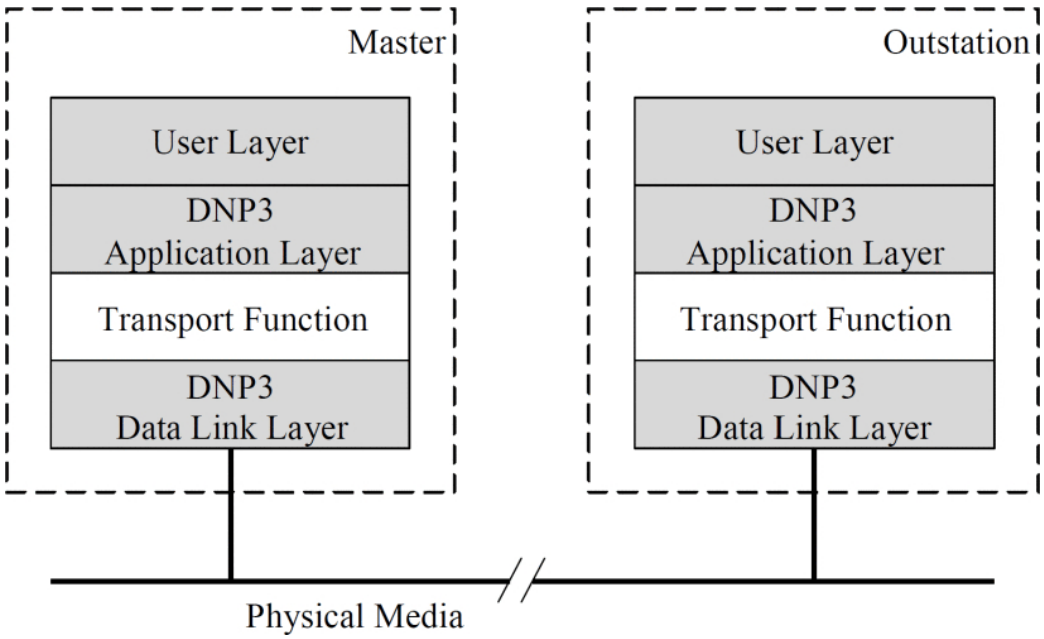

Figure 5.4: DNP3 Network Stack [14].

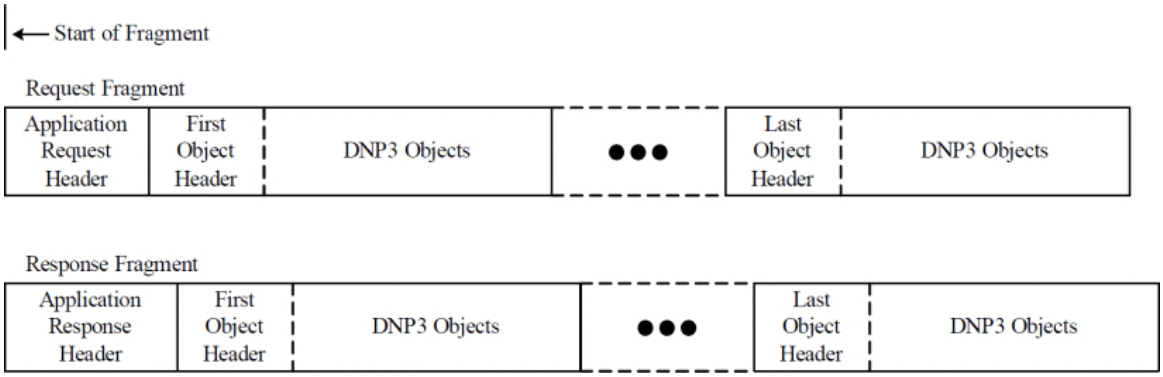

Figure 5.5: DNP3 Fragment structure [14].

layer and transport function [36]. This layer is responsible for providing station addressing and error detection functionalities. The data link layer makes no assumptions about the physical communication medium used. It simply assumes that the data is transmitted and received as a continuous stream[14]. Because of this, the physical layer underneath can be both a connection-less or a connection-oriented system. As an example, both TCP/IP and UDP/IP are supported and viewed simply as a generic data stream by the data link layer.

DNP3 fragments produced by the application layer are wrapped in data link frames that are created by the data link layer [36]. Figure 5.6 illustrates the format. Each frame start with a fixed length header block that is followed by zero or more data blocks. The header

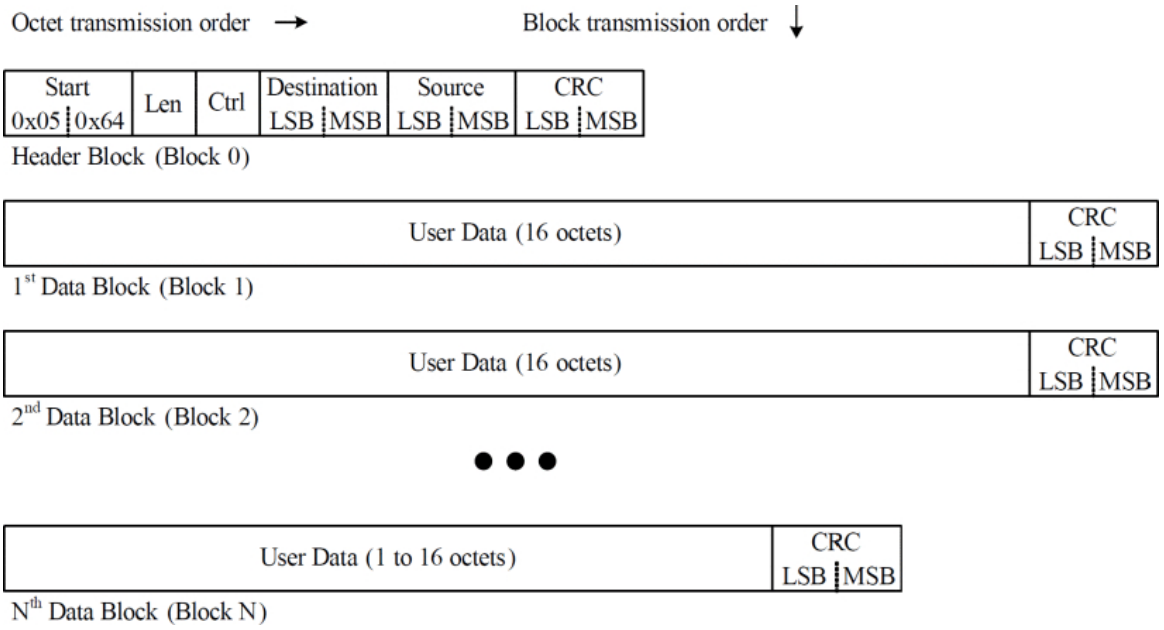

Figure 5.6: DNP3 Data Link Frame [14].

and each data block end with a 16-bit cyclic redundancy check (CRC). The following fields are contained in the header: the *start field*, the *length field*, the *control field*, the *destination address field*, the *source address fields*, and the *CRC field* [14]. The start field is used to mark the beginning of a frame. The length field denotes the size of the following user data. The control field relays information about the frame type as well as flow control. The address fields contain the source and destination MAC addresses. The CRC field contains the CRC check code.

5.3 Protocol comparisons

Table 5.2 presents a brief listing of some comparable attributes of few SCADA communication protocols. Out of the listed protocols, the open source DNP3, IEC 60870-5-101 and Foundation Fieldbus are widely used [47]. Out of DNP3 and IEC 60870-5-101, DNP3 has larger packet sizes which makes it more suitable option for longer distances. Unlike DNP3 and IEC 60780 that use four and tree layer architectures respectively, Modbus is a single layer protocol. It is best suited for situations where data volumes are low. Modbus does not support encryption nor authentication control. Hence, its only viable in deployments where communication security is ensured via other means [8].

Name	Year	Organization	# of layers	Architecture	Addressing scheme	Used in	Source	Security state
Modbus	1973	Gould Modicon	1	Single layer i.e. Application layer	8-bit addresses	Targets low volume data applications	Open source	No encryption or authentication control
DNP3	1993	Harris, Distributed Automation Products	4	4-layer architecture	16-bit addresses	China, North America, and Australia	Open source	DNP3-SA supports encryption and authentication control
IEC 6870-5-101	1995	IEC	3	3-layer architecture based on EPA model.	Support for 0, 8, and 16-bit addresses	Europe, China	Commercially available	No encryption, supports authentication control
Foundation Fieldbus	2004	FieldComm Group	4	4-layer architecture	Support for 8, 16, and 32-bit addresses	America and France	Open source	No encryption or authentication control
Profibus	1989	Promoted by BMBF (Germany)	3	3-layer architecture	7-bit addresses	All over the world	Commercially available	Support for encryption and authentication control
IEC 61850	2005	IEC Technical Committee 57	3	3-layer architecture	48-bit addresses	All over the world	Open source	No encryption, supports authentication control

Table 5.2: Comparison of various communication protocol features. Amended from [47].

6 SCADA communication security

Sufficiently hardened communication protocols and overall communication security play a pivotal role in ensuring that SCADA systems can operate uninterrupted. This chapter focuses on SCADA communication security. The chapter starts with an overview of a classification scheme that helps to map different attack types to their targeted system security properties. Lastly, some SCADA communication security standards and the parties publishing them are presented.

6.1 Overview

SCADA systems have been increasingly targeted by cyber attacks since standardization efforts have introduced common IT networking technologies and protocols into them [12]. Increased Internet connectivity and the usage of wireless communication technologies have increased the attack surface of SCADA systems [48]. This coupled with the lack of strong encryption schemes and real-time monitoring has made reconnaissance and attack planning more straightforward for malicious actors [2].

The nature of SCADA systems make them a high value target for cybercriminals and nation-state actors [34]. Critical infrastructure such as power plants and water treatment facilities are controlled and monitored by SCADA systems. Thus, SCADA systems are a prime target for an attacker attempting to cause disruption or damage.

Particularly, the introduction of Ethernet, TCP/IP, and wireless technologies such as IEEE 802.x and Bluetooth into SCADA networks has significantly increased connectivity to other networks and thus reduced isolation [38]. This ability to interconnect with other networks also means that network attacks launched for interconnected networks can reach SCADA systems and escalate into an attack against it.

Attacks against SCADA networks communication stack can happen on the network, transport, and the application layer of the ISO/OSI communication model. Figure 6.1 presents a classification of some common attack types and maps them to desired system security goals. These security goals are confidentiality, integrity, availability and non-repudiation.

Security goals for control systems, such as SCADA, weigh these security goals slightly

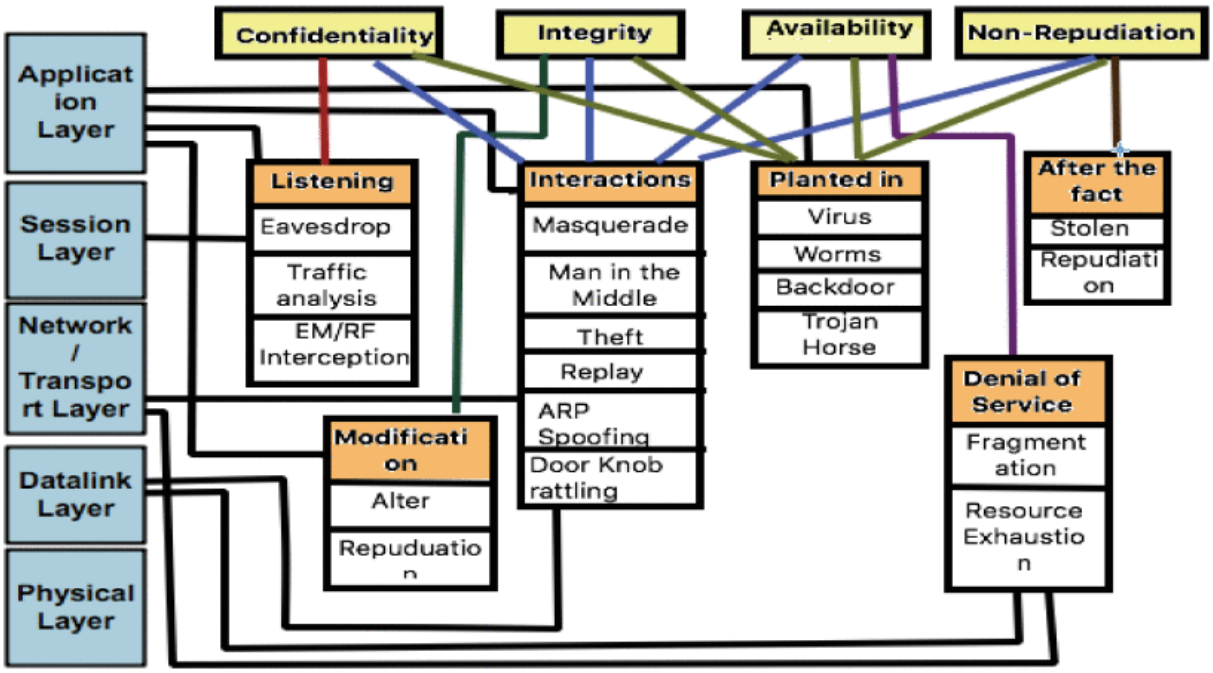

Figure 6.1: Classification of attacks against SCADA networks [12].

different compared to standard IT systems [2, 34]. Availability can be further broken down to responsiveness. A command given to a field device needs to get executed within a hard time constraint. Confidentiality of the system's critical information such as encryption keys, passwords, and system layout is paramount. However, the message confidentiality of control and data messages is not of vital importance. This emerges as consequence of the processes that SCADA systems control. They are continuous physical processes and the nature of commands and responses used to control the process is repetitive and relatively predictable. Data transmitted in a SCADA system needs to have high integrity [38]. This expands to cover both the content and the header of data packages. Data payload integrity alone is not sufficient. Perfectly legitimate control commands can cause havoc if they are redirected to wrong field device.

Figure 6.1 groups different attack types into six different groups. Each group is then linked into corresponding OSI layers and security properties. Listening type of attacks are linked to session and application layer and they are mapped to the confidentiality security property. Modification type of attacks are linked to application layer and are mapped to integrity security property. Fact altering attacks have no direct link to any OSI layer but they are mapped to non-repudiation property. Interaction type of attacks are linked to datalink, transport, and application layers and they are mapped to all four system security properties. Denial of Service (DoS) attacks are linked to datalink and

transportation layerss and are mapped to availability property. Lastly, planted in type of attacks where some form of executable such as a worm or a virus is used is linked to application layer and mapped to all four system security properties.

Some attack vectors that leverage the increased connectivity of SCADA systems are listed in [11]. These attack vectors are *modems*, *wireless networks*, *third-party connections*, *VPNSs*, *Mobile devices*, and lastly, *Internet* itself. Most modems lack strong passwords. They can even have none set or use a weak one. Wireless networks make it possible for malicious actors to spoof themselves as legitimate network nodes. They can then attempt to communicate with the control center. Remote support systems that initiate third-party connections can open attack surface to the system and thus make the system more accessible for malicious actors. When implemented correctly, VPNs do not expose extra attack surface. However, their security level can pose major threat to system safety. Mobile devices bring with them an external ecosystem of various security policies and practices. If connected mobile devices are not sufficiently hardened they can expose additional attack surface. Lastly, with increased Internet connectivity a SCADA system gets increasingly accessible for remotely operating malicious actors.

6.2 Securing SCADA communication standards

There are several national and international standards that can be used to harden OT systems. These standards provide guidelines and recommendations.

The very first SCADA systems used in the late 1960's collected telemetry data for power grids [5]. Communication protocols deployed for these early SCADA systems had no security mechanisms in place. Security standard development for SCADA protocols kicked off as late as 2007, with the release of the first edition of *IEC standard 62351* [19], a standard detailing the security of data exchange with the IEC 61850 [18] protocols. This was followed by the release of the security extension *IEC TS 60870-5-7:2013* [26] in 2013. The extension contains security standards for IEC 60870 [17] protocol. *IEC 62443* [25] standard was published in 2021. The standard contains security best practices covering technologies from both OT and IT systems used in ICS.

Some securing SCADA communication standards released by the International Electrotechnical Commission (IEC) are listed below [11]:

1. IEC 62351-1 [27]: and IEC 62351-2 [28]: contain glossary as well as a formal intro-

duction to existing SCADA communication security standards.

2. IEC 62351-3 [20]: Data and Communication Security: contains security standards for profiles utilizing TCP/IP. It specifies transport-level security in regards to confidentiality and authentication.
3. IEC 62351-4 [21]: Data and Communication Security - Profiles: contains security standards to profiles using ISO-9506 [30] manufacturing message specification. It allows both secure and nonsecure communications to use transport-level security.
4. IEC 62351-5 [22]: Data and Communication Security discusses SCADA communication protocols IEC 60870-5 and its add-ons.
5. IEC 62351-6 [23] and 62351-7 [24] provide standards for peer-to-peer protocols and network system management security.
6. IEC 62443 [25] series offer guidelines for securing ICS and OT systems.

Organizations and companies can ensure that they are following industry best practices and complying with regulatory requirements by adopting appropriate standards [34].

Some standards published by other parties are listed below:

1. *NIST SP 800-82* [41] published by the National Institute of Standards and Technology (NIST). Contains recommendations for securing ICS and OT systems on topics covering network security, access control and incident response.
2. *ISO 27001* [29] standard describes a framework for information security managements systems (ISMS) that also includes guidelines for securing OT systems.
3. *ENISA's OT Cybersecurity Recommendations*[*] are published by the European Union Agency for Cybersecurity (ENISA). These publications offer recommendations on how to harden OT systems over topics of threat intelligence, network security and incident response. For an example document published by ENISA, see e.g., *Communication network dependencies for ICS/SCADA systems*, *ENISA*; 2017. [10].

[*] https://www.enisa.europa.eu/publications

7 Discussion

This thesis examined SCADA systems and their security aspects, especially from the point of view of communication security.

There are various types of operational technology (OT) systems. An OT system consists of two distinct parts: process and control. SCADA is an ubiquitous framework used in modern industrial control systems for monitoring and controlling OT equipment. SCADA systems saw their first iteration in the 1970's and has since evolved from its initial monolithic architecture into complex, networked systems.

One major factor differentiating OT systems from IT systems is that any logic execution causes direct, measurable impact in the physical world. However, security weaknesses of SCADA do not radically differ from those of traditional IT systems. Especially the adaptation of modern networking technologies has brought with it interconnectivity and various security implications for SCADA.

The desired security properties in IT-systems, known as the CIA triad, standing for confidentiality, integrity, and availability apply for SCADA systems as well. However, in SCADA, a special emphasis is placed on system availability and data integrity. Should the system fail to uphold either one, it can endanger human and plant safety.

On a high level, SCADA systems consists of a control center that is connected to a number of field sites through a communication medium. The architecture of these systems has evolved from the first generation systems into modern fourth generation systems. The first generation of SCADA architectures used solely proprietary communication protocols and used fieldbus networks as communication mediums. They were thus heavily dependant on a single vendor. Upcoming fourth generation architecture, on the other hand is heavily cloud-based and utilizes Industrial Internet of Things (IIoT) and Software Defined Networking (SDN) technologies.

SCADA systems are hard real-time systems that have to meet the given time constraints to ensure safety of the controlled physical processes. Field devices that are used to operate valves and actuators on field sites are embedded systems with limited computing resources. These field devices are required to operate continuously with as little downtime as possible. This causes challenges for the lifecycle management and patching of these devices.

Legacy SCADA systems are not compatible with the typical modern hardening methods. The fundamental building blocks of contemporary system security, such as encryption and authentication cannot be implemented just as they are into legacy systems. There are, however, some defensive non-intrusive add-on mechanisms that can be implemented to provide such features.

Modern SCADA systems utilize internet technologies. The introduction of Ethernet-based networks into SCADA systems made it possible to leverage contemporary IT technologies for network communication. Also, old proprietary protocols are less frequently used and more SCADA systems run open source protocols. SCADA systems are designed with longer lifetimes than IT networks. This difference introduces some amount of technical debt as SCADA systems have to offer longer backward compatibility support.

The discrepancy between the expected lifetime of IT and SCADA systems poses challenges to SCADA communication security. Remotely operating malicious actors can connect to SCADA systems from adjacent corporate networks and weaponize legacy vulnerabilities present in older SCADA communication protocols. The increased interconnectivity from and to public Internet and the usage of IT technologies have made SCADA systems suspect to most attacks and threats facing any Internet-connected IT system. This matter is further complicated by the fact that modern SCADA systems are large and can contain several hundreds of thousands of fields devices, each of which is a potential entry point for malicious actors.

Nowadays more and more standards, frameworks and documented industry best practices for securing SCADA systems exist. Most are openly accessible and published by standard and regulatory bodies such as ISO, NIST and ENISA. As usually with such standards, implementing them is highly recommendable as they help prune out the majority of low-hanging fruits from the reach of malicious actors. An increasing amount of literature document known cyber incidents affecting ICS and SCADA systems. One such, well-known incident took place in Ukraine in December 2015 when the Advanced Persistent Threat (APT) group *Sandworm* used malware named *BlackEnergy 3* to cause wide-spread power outages affecting over two hundred thousand Ukrainians. This incident is just one of the many examples of how SCADA systems are increasingly targeted by malicious actors. Thus, keeping up to date with security standards and frameworks is more important than ever.

This thesis provided an overview of the main focal areas of SCADA systems and their security aspects. A deeper look was taken especially into SCADA protocols and com-

munication security. More and more SCADA systems are either directly or indirectly connected to the public Internet. This means that an increasing number of SCADA systems are discoverable directly or indirectly from the Internet. This also means that an increasing number of SCADA systems will be getting probed by various kinds of automatic scanners deployed by malicious actors such as ransomware gangs. These scanners search for any exploitable, unpatched system components. The inherent long patching cycles and the common use of legacy components in SCADA systems can expose them to such threats which previously plagued only IT systems.

For this reason one recommendable focus of further study could be looking into what kind of, if any, methods there exists in the literature to combat these issues brought by the increased Internet presence of SCADA systems. Any such methods should take into account the characteristic of SCADA systems, notably the fact that they host numerous embedded devices running legacy software with limited computing capabilities.

Bibliography

[1] H. Abbas. "Future SCADA Challenges and the Promising Solution: The Agent-Based SCADA". In: *International Journal of Critical Infrastructures* 10.3-4 (2014), pp. 307–333. ISSN: 1475-3219. DOI: 10.1504/IJCIS.2014.066354.

[2] S. D. D. Anton, D. Fraunholz, D. Krohmer, D. Reti, D. Schneider, and H. D. Schotten. "The Global State of Security in Industrial Control Systems: An Empirical Analysis of Vulnerabilities Around the World". In: *IEEE Internet of Things Journal* 8.24 (Dec. 2021), pp. 17525–17540. ISSN: 2327-4662. DOI: 10.1109/JIOT.2021.3081741.

[3] BEUG. *BITBUS specification overview*. http://www.bitbus.org/fprimer.htm. BITBUS European Users Groug e.V., 2003.

[4] J. H. Castellanos, D. Antonioli, N. O. Tippenhauer, and M. Ochoa. "Legacy-Compliant Data Authentication for Industrial Control System Traffic". In: *Applied Cryptography and Network Security*. Ed. by D. Gollmann, A. Miyaji, and H. Kikuchi. Springer International Publishing, 2017, pp. 665–685. ISBN: 978-3-319-61204-1. DOI: 10.1007/978-3-319-61204-1_33.

[5] A. C.-F. Chan and J. Zhou. "Non-Intrusive Protection for Legacy SCADA Systems". In: *IEEE Communications Magazine* 61.6 (June 2023), pp. 36–42. ISSN: 1558-1896. DOI: 10.1109/MCOM.003.2200564.

[6] A. C.-F. Chan and J. Zhou. "Toward Safe Integration of Legacy SCADA Systems in the Smart Grid". In: *Applied Cryptography and Network Security Workshops*. Ed. by J. Zhou, S. Adepu, C. Alcaraz, L. Batina, E. Casalicchio, S. Chattopadhyay, C. Jin, J. Lin, E. Losiouk, S. Majumdar, W. Meng, S. Picek, J. Shao, C. Su, C. Wang, Y. Zhauniarovich, and S. Zonouz. Springer International Publishing, 2022, pp. 338–357. ISBN: 978-3-031-16815-4. DOI: 10.1007/978-3-031-16815-4_19.

[7] P. Church, H. Mueller, C. Ryan, S. V. Gogouvitis, A. Goscinski, H. Haitof, and Z. Tari. "SCADA Systems in the Cloud". In: *Handbook of Big Data Technologies*. Ed. by A. Y. Zomaya and S. Sakr. Cham: Springer International Publishing, 2017, pp. 691–718. DOI: 10.1007/978-3-319-49340-4_20.

[8] C. Davidson, T. Andel, M. Yampolskiy, J. Mcdonald, W. Glisson, and T. Thomas. "On SCADA PLC and Fieldbus Cyber-Security". In: *13th International Conference on Cyber Warfare and Security*. Mar. 2018, pp. 140–149.

[9] M. B. de Freitas, L. Rosa, T. Cruz, and P. Simões. "SDN-Enabled Virtual Data Diode". In: *Computer Security*. Ed. by S. K. Katsikas, F. Cuppens, N. Cuppens, C. Lambrinoudakis, A. Antón, S. Gritzalis, J. Mylopoulos, and C. Kalloniatis. Springer International Publishing, 2019, pp. 102–118. ISBN: 978-3-030-12786-2. DOI: 10.1007/978-3-030-12786-2_7.

[10] European Union Agency for Cybersecurity. *Communication network dependencies for ICS/SCADA systems*. European Network and Information Security Agency (ENISA), Feb. 2017. ISBN: 978-92-9204-192-2. DOI: 10.2824/397676.

[11] J. Gao, J. Liu, B. Rajan, R. Nori, B. Fu, Y. Xiao, W. Liang, and C. L. Philip Chen. "SCADA Communication and Security Issues". In: *Security and Communication Networks* 7.1 (2014), pp. 175–194. ISSN: 1939-0122. DOI: 10.1002/sec.698.

[12] S. Ghosh and S. Sampalli. "A Survey of Security in SCADA Networks: Current Issues and Future Challenges". In: *IEEE Access* 7 (2019), pp. 135812–135831. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2019.2926441.

[13] A. Homay, C. Chrysoulas, B. E. Boudani, M. de Sousa, and M. Wollschlaeger. "A Security and Authentication Layer for SCADA/DCS Applications". In: *Microprocessors and Microsystems* 87 (Nov. 2021), p. 103479. ISSN: 0141-9331. DOI: 10.1016/j.micpro.2020.103479.

[14] IEEE. "IEEE Standard for Electric Power Systems Communications-Distributed Network Protocol (DNP3)". In: *IEEE Std 1815-2012 (Revision of IEEE Std 1815-2010)* (Oct. 2012), pp. 1–821. DOI: 10.1109/IEEESTD.2012.6327578.

[15] V. M. Igure, S. A. Laughter, and R. D. Williams. "Security Issues in SCADA Networks". In: *Computers & Security* 25.7 (Oct. 2006), pp. 498–506. ISSN: 0167-4048. DOI: 10.1016/j.cose.2006.03.001.

[16] M. Ike, K. Phan, K. Sadoski, R. Valme, and W. Lee. "Scaphy: Detecting Modern ICS Attacks by Correlating Behaviors in SCADA and PHYsical". In: *2023 IEEE Symposium on Security and Privacy (SP)*. May 2023, pp. 20–37. DOI: 10.1109/SP46215.2023.10179411.

[17] International Electrotechnical Commission (IEC). *IEC 60870-5 Series – Telecontrol equipment and systems – Transmission protocols*. Standard IEC 60870-5:2024 SER. Technical Committee 57 (TC 57). 2024. URL: https://webstore.iec.ch/publication/3755.

[18] International Electrotechnical Commission (IEC). *IEC 61850 Series – Communication Networks and Systems in Substations*. Standard IEC 61850:2024 SER. Technical Committee 57 (TC 57). 2024. URL: https://webstore.iec.ch/publication/6028.

[19] International Electrotechnical Commission (IEC). *IEC 62351 Series – Power systems management and associated information exchange - Data and communications security*. Standard IEC 62351:2007 SER. Technical Committee 57 (TC 57). 2007. URL: https://webstore.iec.ch/publication/6912.

[20] International Electrotechnical Commission (IEC). *IEC 62351-3:2023 – Communication network and system security - Profiles including TCP/IP*. Standard IEC 62351-3:2023. Technical Committee 57 (TC 57). 2023. URL: https://webstore.iec.ch/publication/68410.

[21] International Electrotechnical Commission (IEC). *IEC 62351-4:2018 – Communication network and system security - Profiles including MMS and Similar Payloads*. Standard IEC 62351-4:2018. Technical Committee 57 (TC 57). 2018. URL: https://webstore.iec.ch/publication/67350.

[22] International Electrotechnical Commission (IEC). *IEC 62351-5:2023 – Security for IEC 60870-5 and derivatives*. Standard IEC 62351-5:2023. Technical Committee 57 (TC 57). 2023. URL: https://webstore.iec.ch/publication/65511.

[23] International Electrotechnical Commission (IEC). *IEC 62351-6:2020 – Security for IEC 61850 Peer-to-Peer Profiles*. Standard IEC 62351-6:2020. Technical Committee 57 (TC 57). 2023. URL: https://webstore.iec.ch/publication/63742.

[24] International Electrotechnical Commission (IEC). *IEC 62351-7:2017 – Network and System Management (NSM) data object models*. Standard IEC 62351-7:2017. Technical Committee 57 (TC 57). 2017. URL: https://webstore.iec.ch/publication/30593.

[25] International Electrotechnical Commission (IEC). *IEC 62443 Series – Security for industrial automation and control systems*. Standard IEC 62443:2023 SER. Technical Committee 65 (TC 65). 2023. URL: https://www.isa.org/standards-and-publications/isa-standards/isa-iec-62443-series-of-standards.

[26] International Electrotechnical Commission (IEC). *IEC TS 60870-5-7:2013 – Telecontrol equipment and systems – Part 5-7: Transmission protocols*. Standard IEC TS 60870-5-7:2013. Technical Committee 57 (TC 57). 2013. URL: https://webstore.iec.ch/publication/3754.

[27] International Electrotechnical Commission (IEC). *IEC TS 62351-1:2007 – Introduction to the standard*. Standard IEC TS 62351-1:2007. Technical Committee 57 (TC 57). 2007. URL: https://webstore.iec.ch/publication/6903.

[28] International Electrotechnical Commission (IEC). *IEC TS 62351-2:2008 – Glossary of terms*. Standard IEC TS 62351-2:2008. Technical Committee 57 (TC 57). 2008. URL: https://products.iec.ch/view/pub/6905.

[29] International Organization for Standardization (ISO). *ISO/IEC 27001:2022 – Information security, cybersecurity and privacy protection – Information security management systems – Requirements*. Standard ISO/IEC 27001:2022. Oct. 2022. URL: https://www.iso.org/standard/27001.

[30] International Organization for Standardization (ISO). *ISO/IEC 9506:2003 – Industrial automation systems – Manufacturing Message Specification*. Standard ISO/IEC 9506:2003. Aug. 2003. URL: https://www.iso.org/standard/37079.html.

[31] S. Jaloudi. "Communication Protocols of an Industrial Internet of Things Environment: A Comparative Study". In: *Future Internet* 11.3 (Mar. 2019), p. 66. ISSN: 1999-5903. DOI: 10.3390/fi11030066.

[32] R. Langner. "Stuxnet: Dissecting a Cyberwarfare Weapon". In: *IEEE Security & Privacy* 9.3 (May 2011), pp. 49–51. ISSN: 1558-4046. DOI: 10.1109/MSP.2011.67.

[33] S. McLaughlin, C. Konstantinou, X. Wang, L. Davi, A.-R. Sadeghi, M. Maniatakos, and R. Karri. "The Cybersecurity Landscape in Industrial Control Systems". In: *Proceedings of the IEEE* 104.5 (May 2016), pp. 1039–1057. ISSN: 1558-2256. DOI: 10.1109/JPROC.2015.2512235.

[34] M. Mesbah, M. S. Elsayed, A. D. Jurcut, and M. Azer. "Analysis of ICS and SCADA Systems Attacks Using Honeypots". In: *Future Internet* 15.7 (July 2023), p. 241. ISSN: 1999-5903. DOI: 10.3390/fi15070241.

[35] S. Nazir, S. Patel, and D. Patel. "Assessing and Augmenting SCADA Cyber Security: A Survey of Techniques". In: *Computers & Security* 70 (Sept. 2017), pp. 436–454. ISSN: 0167-4048. DOI: 10.1016/j.cose.2017.06.010.

[36] D. Njova, K. Ogudo, and P. Umenne. "Modelling the IEC 61850 and DNP3 Protocol Using OPNET in an Electrical Substation Communication Network". In: *2022 International Conference on Artificial Intelligence, Big Data, Computing and Data Communication Systems (icABCD)*. Aug. 2022, pp. 1–7. DOI: 10.1109/icABCD54961.2022.9856151.

[37] A. Nourian and S. Madnick. "A Systems Theoretic Approach to the Security Threats in Cyber Physical Systems Applied to Stuxnet". In: *IEEE Transactions on Dependable and Secure Computing* 15.1 (Jan. 2018), pp. 2–13. ISSN: 1941-0018. DOI: 10.1109/TDSC.2015.2509994.

[38] D. Pliatsios, P. Sarigiannidis, T. Lagkas, and A. G. Sarigiannidis. "A Survey on SCADA Systems: Secure Protocols, Incidents, Threats and Tactics". In: *IEEE Communications Surveys & Tutorials* 22.3 (2020), pp. 1942–1976. ISSN: 1553-877X. DOI: 10.1109/COMST.2020.2987688.

[39] L. C. Ruiz Salvador, N. Huu Phuoc Dai, and R. Zoltán. "SCADA Systems: Security Concerns and Countermeasures". In: *2023 IEEE 21st World Symposium on Applied Machine Intelligence and Informatics (SAMI)*. Jan. 2023, pp. 000251–000254. DOI: 10.1109/SAMI58000.2023.10044495.

[40] T. Sauter. "The Three Generations of Field-Level Networks—Evolution and Compatibility Issues". In: *IEEE Transactions on Industrial Electronics* 57.11 (Nov. 2010), pp. 3585–3595. ISSN: 1557-9948. DOI: 10.1109/TIE.2010.2062473. (Visited on 06/01/2024).

[41] K. Stouffer, M. Pease, C. Tang, T. Zimmerman, V. Pillitteri, S. Lightman, A. Hahn, S. Saravia, A. Sherule, and M. Thompson. *Guide to Operational Technology (OT) Security*. Tech. rep. NIST Special Publication (SP) 800-82 Rev. 3. National Institute of Standards and Technology, Sept. 2023. DOI: 10.6028/NIST.SP.800-82r3.

[42] M. Sverko, T. G. Grbac, and M. Mikuc. "SCADA Systems With Focus on Continuous Manufacturing and Steel Industry: A Survey on Architectures, Standards, Challenges and Industry 5.0". In: *IEEE Access* 10 (2022), pp. 109395–109430. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2022.3211288.

[43] I. Verhappen and A. Pereira. *Foundation Fieldbus*. 3rd edition. ISA, Feb. 2009. ISBN: 978-1-934394-76-2.

[44] A. Volkova, M. Niedermeier, R. Basmadjian, and H. de Meer. "Security Challenges in Control Network Protocols: A Survey". In: *IEEE Communications Surveys & Tutorials* 21.1 (2019), pp. 619–639. ISSN: 1553-877X. DOI: 10.1109/COMST.2018.2872114.

[45] M. Wollschlaeger, T. Sauter, and J. Jasperneite. "The Future of Industrial Communication: Automation Networks in the Era of the Internet of Things and Industry 4.0". In: *IEEE Industrial Electronics Magazine* 11.1 (Mar. 2017), pp. 17–27. ISSN: 1941-0115. DOI: 10.1109/MIE.2017.2649104.

[46] A. K. Wright, J. A. Kinast, and J. McCarty. "Low-Latency Cryptographic Protection for SCADA Communications". In: *Applied Cryptography and Network Security*. Ed. by M. Jakobsson, M. Yung, and J. Zhou. Springer, 2004, pp. 263–277. ISBN: 978-3-540-24852-1. DOI: 10.1007/978-3-540-24852-1_19.

[47] G. Yadav and K. Paul. "Architecture and Security of SCADA Systems: A Review". In: *International Journal of Critical Infrastructure Protection* 34 (Sept. 2021), p. 100433. ISSN: 1874-5482. DOI: 10.1016/j.ijcip.2021.100433.

[48] B. Zhu, A. Joseph, and S. Sastry. "A Taxonomy of Cyber Attacks on SCADA Systems". In: *2011 International Conference on Internet of Things and 4th International Conference on Cyber, Physical and Social Computing*. Oct. 2011, pp. 380–388. DOI: 10.1109/iThings/CPSCom.2011.34.

[49] B. Zhu, A. Joseph, and S. Sastry. "A Taxonomy of Cyber Attacks on SCADA Systems". In: *2011 International Conference on Internet of Things and 4th International Conference on Cyber, Physical and Social Computing*. Oct. 2011, pp. 380–388. DOI: 10.1109/iThings/CPSCom.2011.34.